



**LO5 WISSENSCHAFTLICHE
OPEN-ACCESS-
RESSOURCEN: DIGITALE
DATENBANKEN**

Fortgeschrittenes Niveau

AUTOR:

FABIANO CHALHOUB & ZOI GEORGIU



Inhalt

Erweiterte Struktur einer Datenbank	4
Erweiterte Struktur einer Datenbank.....	4
Datenbankmanagementsystem	4
Sprachen der Datenbankmodelle.....	4
Datenbankeigenschaften	5
Datenbankmodelltypen	5
Phasen des Datenbankdesigns.....	6
Abstraktionsgrad	7
Datenbankschemata.....	7
Externe Ebene	7
Konzeptionelle Ebene	7
Interne Ebene	8
Datenunabhängigkeit.....	8
Das relationale Datenmodell	8
Grundelemente eines relationalen Datenmodells	9
Eigenschaften einer Tabelle	9
Unterscheidungsmerkmale des relationalen Datenbankmodells	10
Das Entity-Relationship-Modell	10
Beispiele für Entitätstypen und Beziehungen in biologischen Datenbanken.....	11
Modifikationsanomalien	12
Schlüsseldefinitionen	12
Strukturierte Abfragesprache (SQL).....	13
Liste der SQL-Befehle	13
Kommerzielle und kostenlose Datenbanken, die in der realen Welt verwendet werden	18
Kommerzielle Datenbanken.....	18
SAP HANA.....	19
IBM Db2-Datenbank.....	19
Oracle-Datenbank	20



2019-1-BG01-KA203-062371

Kostenlose Datenbanken	21
MySQL.....	21
PostgreSQL	22
Microsoft SQL	22
MariaDB.....	23
Orakel.....	23
Firebirdsql	24
Datenbanken in der wissenschaftlichen Welt Aufbaumodul	24
Übersicht über Datenbanken in der wissenschaftlichen Welt	24
DNA-Datenbanken.....	26
RNA-Datenbanken	26
Proteindatenbanken	27
Krankheitsdatenbanken	27
Ausdrucksdatenbanken.....	27
Pfaddatenbanken	27
Die Datenbank des Medical Information Mart for Intensive Care (MIMIC).....	30
PCORnet	31
NHS öffnen	31
De-Identifikation der Datenbank.....	31
Die Anwendung von Blockchain in der digitalen Gesundheit.....	32
Verweise.....	34



Erweiterte Struktur einer Datenbank

ERWEITERTE STRUKTUR EINER DATENBANK

Dieser Teil befasst sich mit dem fortgeschrittenen Design einer Datenbank. Es erklärt die Struktur einer Datenbank und wie man Beziehungen zwischen Datenbanktabellen herstellt. Es präsentiert auch die spezifische Sprache, die verwendet wird, um Abfragen (SQL) zu erstellen, um Daten aus einer Datenbank abzurufen.

DATENBANKMANAGEMENTSYSTEM

Eine moderne Datenbank kann als strukturierte Sammlung von Informationen (Daten) definiert werden, die für die reale Welt repräsentativ sind. Database Management Systeme (DBMS) dienen der Erstellung, Verwaltung und Abfrage von Datenbanken. Derzeit sind relationale Datenbankmanagementsysteme (RDBMS) die ausgereiftesten und am weitesten verbreiteten Datenbanksysteme in der Produktion. Fast alle Online-Transaktionen und die meisten Online-Content-Management-Systeme (zB Blogs und soziale Netzwerke) basieren auf solchen Datenbanksystemen, die für die Anwendungsinfrastruktur der Welt von zentraler Bedeutung sind. Der Schwerpunkt eines DBMS ist die Zusammenstellung von Diensten, die die Persistenz von Daten in der Datenbank und die Funktionalität bieten, um sicherzustellen, dass die Daten korrekt und konsistent sind und Transaktionen den ACID-Eigenschaften folgen. ACID bezieht sich auf vier wesentliche Eigenschaften einer Transaktion:

- Atomarität
- Konsistenz
- Isolation
- Haltbarkeit

SPRACHEN DER DATENBANKMODELLE

Alle Datenbankmodelle verfügen über eine Sprache zur Spezifikation von Struktur und Inhalt der Datenbank. Die Spezifikation wird als Schemadesign bezeichnet und stellt die logische Sicht auf Informationen dar, die von einem bestimmten DBMS verwaltet werden. Diese Datenbankspezifikationsprache muss flexibel sein, damit sie nützlich und dauerhaft ist. Das sichtbarste Element einer Datenbank, das von Datenbankexperten und Anwendungsentwicklern identifiziert werden kann, ist die Datenbearbeitungssprache. Es kann viele Formen aufweisen, wobei die häufigste eine programmiersprachenähnliche Schnittstelle ist. Heute sind die Text- und Prozedursprachen,



2019-1-BG01-KA203-062371

einschließlich der Structured Query Language (SQL) und der Object Query Language (OQL), die am weitesten verbreiteten Formen der Datenbearbeitungssprache.

DATENBANKEIGENSCHAFTEN

Eine Datenbank kann als kohärent, logisch und intern konsistent charakterisiert werden. Es kann auch als selbstbeschreibend bezeichnet werden, da es Metadaten enthält, die die Daten und Beziehungen zwischen Tabellen in der Datenbank definieren und beschreiben. Es ist so konzipiert, dass es Daten für einen bestimmten Zweck enthält. Jedes Datenelement wird in einem Feld gespeichert; eine Kombination von Feldern wird als Tabelle bezeichnet. In einer Datenbank können mehrere Tabellen vorhanden sein.

Im Gegensatz zum dateibasierten System wird bei Datenbanksystemen die Datenstruktur im Systemkatalog und nicht in den Anwendungsprogrammen abgelegt. Diese Trennung zwischen Programmen und Daten wird als Programm-Daten-Unabhängigkeit bezeichnet.

Die Architektur eines Datenbanksystems besteht aus einem Satz von Diensten, die auf Basis von Betriebssystemdiensten, Systemdateispeicherdiensten und primären Speicherpufferverwaltungsdiensten aufgebaut sind. Dieser Satz von Diensten umfasst Folgendes: Katalogverwaltung, Integritätsverwaltung, Transaktionsverwaltung, Gleichzeitigkeitskontrolle, Sperrverwaltung, Deadlock-Verwaltung, Wiederherstellungsverwaltung, Sicherheitsverwaltung, Abfrageverarbeitung, Kommunikationsverwaltung und Protokollverwaltung.

DATENBANKMODELLTYPEN

Datenmodelle lassen sich in zwei Typen unterteilen:

- Konzeptuelle Datenmodelle auf hoher Ebene
- Datensatzbasierte logische Datenmodelle

Konzeptuelle Datenmodelle auf hoher Ebene schlagen Konzepte für die Präsentation von Daten vor, die der Wahrnehmung von Daten durch Menschen ähneln. Ein Beispiel für dieses Datenmodell ist das Entity-Relationship (ER)-Modell, das auf Konzepten wie Entitäten, Attributen und Beziehungen basiert. Eine Entität entspricht einem realen Objekt, Attribute repräsentieren Eigenschaften der Entität und eine Beziehung zeigt eine Assoziation zwischen Entitäten an.

Datensatzbasierte logische Datenmodelle schlagen Konzepte vor, die Benutzer verstehen können, ähneln jedoch der Art und Weise, wie Daten im Computer gespeichert werden. Relationale Datenmodelle, Netzwerkdatenmodelle und hierarchische Datenmodelle sind drei der am weitesten verbreiteten datensatzbasierten logischen Datenmodelle.



2019-1-BG01-KA203-062371

- Im relationalen Modell werden Daten in Form von Beziehungen oder Tabellen dargestellt.
- Im Netzwerkmodell werden Daten als Datensatztypen dargestellt. Dieses Modell repräsentiert auch einen Mengentyp, der als begrenzter Typ von Eins-zu-Viele-Beziehungen definiert ist.
- In dem hierarchischen Modell werden Daten als eine hierarchische Baumstruktur dargestellt, wobei jeder Zweig davon repräsentativ für eine Anzahl zusammengehöriger Datensätze ist.

PHASEN DES DATENBANKDESIGNS

Die Datenmodellierung ist der erste Schritt des Datenbankdesigns. Dieser Schritt ist jedoch manchmal eine abstrakte Entwurfsphase auf hoher Ebene, die als Konzeptentwurf bekannt ist. In dieser Phase soll Folgendes beschrieben werden:

- Die in der Datenbank vorhandenen Daten
- Die Beziehungen zwischen Datenelementen
- Die Einschränkungen für Daten

In dieser Anfangsphase des Datenbankentwurfsprozesses ist die Analyse des Informationsbedarfs unerlässlich. Dies ist die wichtigste Phase, da die Gesamteffektivität des Systems davon abhängt, wie genau die Informationsanforderungen und Benutzeransichten zu Beginn festgelegt werden. Die in dieser Phase gemachten Angaben zu Informationspflichten wirken sich auf die endgültige Form und den Inhalt des Datenbanksystems aus.

Nachdem die Spezifikationen festgelegt und entwickelt wurden, müssen sie in ein integriertes, zusammenhängendes System strukturiert werden, ein Verfahren namens Logical Design. Das logische Design umfasst die folgenden Schritte:

- i) Entwicklung eines Datenmodells für jede Benutzeransicht
- ii) Integration der Entitäten, Attribute und Beziehungen in ein zusammengesetztes logisches Schema, das die Datenbank für dieses Modul in Begriffen beschreibt, die sich nicht auf das verwendete Softwarepaket beziehen
- iii) Umwandlung des logischen Schemas in ein Softwareschema, das in der Sprache des gewählten Datenbankverwaltungspakets ausgedrückt wird

Der letzte Schritt beim Entwerfen einer Datenbank ist der physische Entwurf. Dieser Schritt ist erforderlich, um das Softwareschema in eine Form zu bringen, die mit der spezifischen Hardware, dem Betriebssystem und dem Datenbankmanagementsystem einer Organisation implementiert werden kann. Das physische Design umfasst die Umsetzung von Integritäts- und Sicherheitsanforderungen sowie das Design von Navigationspfaden.



2019-1-BG01-KA203-062371

ABSTRAKTIONSGRAD

Datenabstraktion bedeutet das Verbergen bestimmter Details der Art und Weise, wie Daten gespeichert und gepflegt werden. Hinsichtlich ihres Abstraktionsgrads lassen sich Datenbankmodelle in drei Ebenen einteilen:

- Die externe oder View-Ebene, die die höchste Abstraktionsebene darstellt und nur einen Teil der gesamten Datenbank darstellt
- Die logische Ebene, die beschreibt, welche Daten in der gesamten Datenbank gespeichert sind
- Die physikalische Ebene, die die unterste Abstraktionsebene ist und beschreibt, wie die Daten in der Datenbank gespeichert werden

DATENBANKSCHEMATA

Das Datenbankschema kann als frühe Datenbankbeschreibung definiert werden, von der nicht erwartet wird, dass sie sich häufig ändert. In einem Datenbanksystem existieren zahlreiche Schemata. Die Datenbankarchitektur besteht aus drei Schemaebenen.

Externe Ebene

Dies ist die höchste Schemaebene. Die Datenansicht auf externer Ebene konzentriert sich auf bestimmte Datenverarbeitungsanwendungen oder Benutzeransichten. Es enthält mehrere Ansichten und stellt ein Fragment der eigentlichen Datenbank dar. Jede Ansicht wird für einen Benutzer oder eine Benutzergruppe angeboten, um die Interaktion zwischen Benutzer und System zu vereinfachen.

Konzeptionelle Ebene

Diese Ebene beschreibt die logische Struktur der gesamten Datenbank, die wiederum durch einfache logische Konzepte beschrieben wird, einschließlich der Objekte, ihrer Eigenschaften oder Beziehungen. Daher ist die Kompliziertheit der Implementierungsdetails der Daten für die Benutzer nicht sichtbar. In der Datenbank wird nur eine Ansicht auf konzeptioneller Ebene verwaltet. Damit auf Entitäten oder Attribute im Datenbanksystem verwiesen werden kann, müssen sie zunächst in der Sicht der konzeptionellen Ebene definiert werden, die formal als logisches Schema bezeichnet wird. Diese Ebenenansicht muss sehr stabil sein, da sie als Grundlage für die Entwicklung von externen und internen Ebenenansichten gilt.



2019-1-BG01-KA203-062371

Interne Ebene

Die Art und Weise der Datenspeicherung und der Zugriff auf die Daten sind in diesem Schema beschrieben. Die interne Ebene repräsentiert den internen oder physischen Zustand der Datenbank. Ihr Ziel ist es, die Effizienz des Datenbanksystems zu steigern und gleichzeitig die erforderlichen Anforderungen zu erfüllen.

DATENUNABHÄNGIGKEIT

Datenunabhängigkeit bezieht sich auf die Fähigkeit von Benutzeranwendungen, von Änderungen in der Definition und Organisation von Daten unberührt zu bleiben. Es gibt zwei Arten von Datenunabhängigkeit: logische und physische.

Logische Datenunabhängigkeit ist die Möglichkeit, das logische (konzeptionelle) Schema zu ändern, ohne das externe Schema oder die Benutzeransicht zu beeinträchtigen. Anpassungen des logischen Schemas, wie zB Änderungen der Datenbankstruktur wie das Hinzufügen von Tabellen, sollten keinen Einfluss auf die Funktion der Anwendung haben (externe Sichten).

Physische Datenunabhängigkeit ist die Fähigkeit des Schemas auf konzeptioneller Ebene, von Änderungen am internen Schema unberührt zu bleiben. Änderungen an Dateioorganisation oder Speicherstrukturen, Speichergeräten oder Indexierungsstrategie bewirken keine Änderungen auf der konzeptionellen Ebene.

DAS RELATIONALE DATENMODELL

Das relationale Datenmodell wurde 1970 von Dr. Edgar F. Codd entwickelt. Es stellt Daten in tabellarischer Form dar, die vielen Menschen vertraut ist. Die logische Einfachheit von Flatfile-Strukturen wird in diesem Modell beibehalten. Das relationale Modell basiert auf einer Mengentheorie, die die Grundlage für mehrere der Operationen bildet, die an Beziehungen durchgeführt werden. Es bietet den flexibelsten Zugriff auf Daten und ist daher in dynamischen Entscheidungsumgebungen nützlich.

SQL ist eine relationale Transformationssprache; es bietet Möglichkeiten, Beziehungen zu bilden und die Daten zu bearbeiten. Das Ergebnis einer Transformationsoperation ist immer eine andere Relation, die nur eine Zeile und eine Spalte enthalten kann.



2019-1-BG01-KA203-062371

Grundelemente eines relationalen Datenmodells

Tabelle 1. Grundkomponenten eines relationalen Datenmodells

Datenbankkomponente	Descrizione
Tabelle	enthält Spalten und Zeilen; eine Teilmenge des kartesischen Produkts einer Liste von Domänen, die durch einen Namen gekennzeichnet sind
Säulen	Hauptspeichereinheiten; enthalten die grundlegenden Datenelemente, in die der Inhalt unterteilt werden kann
Reihen	Spalten enthalten, die verknüpft sind; zusammen mit Spalten bilden die Basis aller Datenbanken
Domain	ein Satz akzeptabler Werte, die in eine Spalte aufgenommen werden können
Grad	die Anzahl der Spalten in einer Tabelle

Eine Relation, die auch als Tabelle oder Datei bezeichnet wird, kann als zweidimensionale Tabelle charakterisiert werden, die aus Daten zu einer Entitätsklasse oder den Beziehungen zwischen Entitätsklassen besteht. In jeder Zeile einer Tabelle sind Daten enthalten, die sich auf eine bestimmte Entität beziehen, und in jeder Spalte ist ein bestimmtes Attribut enthalten. Die Zeilen oder Datensätze einer Relation können als Tupel bezeichnet werden. Ein Datensatz in einer Tabelle repräsentiert eine Instanz einer Entität. Die Anzahl der Zeilen in einer Relation gibt ihre Kardinalität an. Die Anzahl der Spalten, auch Felder oder Attribute genannt, in einer Relation entspricht dem Grad der Relation. Die Grundelemente eines relationalen Datenmodells sind in Tabelle 1 beschrieben. Eine unäre Relation besteht nur aus einem Attribut; eine binäre Relation besteht nur aus zwei Attributen; eine ternäre Relation besteht nur aus drei Attributen.

EIGENSCHAFTEN EINER TABELLE

- Jede Tabelle in einer Datenbank hat einen eindeutigen Namen
- Es sind keine doppelten Zeilen vorhanden; jede reihe ist anders
- Jede Zeile hat einen anderen Namen
- Die Reihenfolge der Zeilen und Spalten ist nicht wichtig
- Einträge aus Spalten werden gemäß ihrem Datentyp aus derselben Domäne abgeleitet, einschließlich: Datum, logisch (wahr/falsch), Zeichen (String) und Zahl (numerisch, Ganzzahl, Gleitkomma, ...)



2019-1-BG01-KA203-062371

UNTERSCHIEDSMERKMALE DES RELATIONALEN DATENBANKMODELLS

Wesentlichkeit: Eine Datenstruktur gilt als wesentlich, wenn sie beim Entfernen zu einem Informationsverlust in der Datenbank führt.

Integritätsregeln: Diese stellen sicher, dass der Datenbankinhalt korrekt und konsistent bleibt. Es gibt zwei Arten von Integrität:

1. **Entitätsintegrität:** Ermöglicht die eindeutige Identifizierung jeder Entität in der relationalen Datenbank. Diese Fähigkeit gewährleistet den Zugriff auf alle Daten. Erfordert, dass kein Primärschlüssel einen Nullwert hat.
2. **Referentielle Integrität:** Ermöglicht die Referenzierung von Tupeln mithilfe von Fremdschlüsseln. Erfordert, dass die von einem Fremdschlüssel angenommenen Werte entweder mit einem in der Datenbank vorhandenen Primärschlüssel übereinstimmen oder vollständig null sind.

Datenmanipulation: Eine Methode zum Manipulieren der Daten; Prinzipieller Ansatz zur Erstellung von Informationen für die Entscheidungsfindung.

DAS ENTITY-RELATIONSHIP-MODELL

Das Entity-Relationship (ER)-Datenmodell ist seit mehr als 35 Jahren verfügbar. Es ist relativ abstrakt und leicht zu erklären. ER-Modelle lassen sich leicht in Beziehungen übersetzen und durch ER-Diagramme darstellen. Beziehungen und Entitäten sind die Grundlagen dieses Modells. Eine Entität kann ein Objekt sein, das physisch existiert oder konzeptionell existiert. Wenn ihre Tabellen existenzabhängig sind, wird eine Entität als schwach bezeichnet. Umgekehrt wird eine Entität als stark bezeichnet, wenn sie getrennt von allen ihren zugehörigen Entitäten existieren kann.

Es gibt verschiedene Arten von Entitäten:

- **Unabhängige Entitäten oder Kernel:** Die Bausteine der Datenbank. Sie sind starke Wesen. Der Primärschlüssel ist kein Fremdschlüssel und kann einfach oder zusammengesetzt sein. Die verschiedenen Schlüsseltypen sind in Tabelle 2 beschrieben.
- **Abhängige oder abgeleitete Entitäten:** Sie sind von zwei oder mehr Tabellen existenzabhängig. Sie werden verwendet, um zwei Kernel zusammenzuführen und können andere Attribute enthalten. Jede verknüpfte Tabelle wird durch den Fremdschlüssel identifiziert. Für den Primärschlüssel stehen drei Optionen zur Verfügung: i) Verwenden Sie einen Verbund aus Fremdschlüsseln verwandter Tabellen, falls eindeutig, ii) Verwenden Sie einen Verbund aus Fremdschlüsseln und einer qualifizierenden Spalte, oder iii) Erstellen Sie einen neuen einfachen Primärschlüssel.



2019-1-BG01-KA203-062371

- Charakteristische Entitäten: Diese Entitäten bieten zusätzliche Informationen zu einer anderen Tabelle. Sie beschreiben andere Entitäten und sind repräsentativ für mehrwertige Attribute. Der Fremdschlüssel dient zur weiteren Identifizierung der charakterisierten Tabelle. Für den Primärschlüssel stehen zwei Optionen zur Verfügung: i) Verwenden eines Verbunds aus Fremdschlüsseln und einer qualifizierenden Spalte oder ii) Erstellen eines neuen einfachen Primärschlüssels.

Tabella 2. Tipi di chiavi

Arten von Schlüsseln	Beschreibung
Kandidatenschlüssel	einfacher oder zusammengesetzter Schlüssel, der eindeutig ist, da keine zwei Zeilen in einer Tabelle zu jeder Zeit denselben Wert haben können, und minimal, da jede Spalte benötigt wird, um Eindeutigkeit zu erreichen
Zusammengesetzter Schlüssel	muss minimal sein; bestehend aus zwei oder mehr Attributen
Primärschlüssel	vom Datenbankdesigner ausgewählter Kandidatenschlüssel zur Verwendung als Identifizierungsmechanismus für den gesamten Entitätssatz; muss Tupel in einer Tabelle eindeutig identifizieren und darf nicht null sein; im ER-Modell durch Unterstreichen des Attributs angezeigt indicated
Sekundärschlüssel	Attribut, das ausschließlich für Abrufzwecke verwendet wird; kann zusammengesetzt sein
Alternativschlüssel	alle Kandidatenschlüssel, die nicht als Primärschlüssel ausgewählt sind
Unbekannter Schlüssel	Attribut in einer Tabelle, die auf den Primärschlüssel in einer anderen Tabelle verweist ODER es kann null sein

Nullwerte: Anders als Null- oder Leerwerte; hängen nicht vom Datentyp ab. Ein Nullwert bedeutet, dass entweder der tatsächliche Wert unbekannt ist oder das Attribut nicht anwendbar ist.

BEISPIELE FÜR ENTITÄTSTYPEN UND BEZIEHUNGEN IN BIOLOGISCHEN DATENBANKEN

Ein Entitätstyp beschreibt die Merkmale, die von einer Sammlung von Entitäten in einer Domäne gemeinsam genutzt werden. Protein kann beispielsweise als Entitätstyp mit Attributen wie Sequenz, Name, Molekulargewicht, Spezies und Zugangsnummer betrachtet werden. Ein einzelner Entitätstyp wird wahrscheinlich mehrere Instanzen haben, von denen jede Werte für die Attribute bereitstellt, die im entsprechenden Typ angegeben sind. Die Namen von zwei Instanzen des Entitätstyps Protein sind beispielsweise menschliches α -Hämoglobin und Walmyoglobin. Die Werte ihrer Attributarten wären jeweils Mensch und Wal.



2019-1-BG01-KA203-062371

Beziehungen geben an, dass zwei oder mehr Entitätstypen verknüpft sind. Zum Beispiel kann ein Protein mit vielen anderen Proteinen interagieren oder kann ein Mitglied einer Familie sein. Verschiedene Kategorien von Beziehungen können die Art der Beziehung beschreiben. Zum Beispiel könnte ein Entitätstyp als Teil eines anderen (zB ein Beta-Strang ist Teil eines Sheets in der Sekundärstruktur eines Proteins) oder als eine Art eines anderen (zB ein Enzym ist eine Art Protein) dargestellt werden.

MODIFIKATIONSANOMALIEN

Während des Einfügens, Löschens oder Ändern von Daten können in einer Datenbank unbeabsichtigte Fehler auftreten. Wenn der Fehler auf das Datenbankdesign zurückzuführen ist, wird dies als Modifikationsanomalie bezeichnet.

Es gibt drei Arten von Modifikationsanomalien:

1. Löschanomalie: die Entfernung einer logischen Entität, die zum Verlust von Informationen über eine nicht verwandte logische Entität führt
2. Einfügeanomalie: das Einfügen von Daten über eine logische Entität, die das Einfügen von Daten über eine nicht verwandte logische Entität erfordert
3. Update-Anomalie: die Änderung der Informationen für eine logische Einheit, die mehr als eine Änderung einer Beziehung erfordert.

SCHLÜSSELDEFINITIONEN

Zentralisiertes Datenbanksystem: Daten in diesem System werden an einem einzigen Standort gespeichert.

Verteiltes Datenbanksystem: Datenbank- und DBMS-Software werden an verschiedenen Standorten verteilt, die durch ein Computernetzwerk verbunden sind.

Datenbank: eine gemeinsame Sammlung zugehöriger Daten zur Unterstützung der Aktivitäten von Organisationen.

Datendefinitionssprache (DDL): verwendet, um die konzeptionellen und internen Schemata zu definieren.

Datenbankverwaltungssystem (DBMS): Computerprogramme zur Erstellung, Verwaltung und Abfrage von Datenbanken.

Datenmodell: eine Sammlung von Konzepten zur Beschreibung der Datenbankstruktur database.



2019-1-BG01-KA203-062371

Daten Redundanz: Speicherung des gleichen Datenstücks an zwei oder mehr Stellen im Datenbanksystem.

Normalisierung: eine Methode, die Daten so strukturiert, dass Probleme verringert oder vermieden werden.

Erholung: das Verfahren zur Verwendung von Protokollen und Sicherungskopien zum Wiederherstellen einer beschädigten Datenbank.

STRUKTURIERTE ABFRAGESPRACHE (SQL)

SQL steht für Structured Query Language, eine Computersprache zum Speichern, Manipulieren und Abrufen von Daten, die in einer relationalen Datenbank gespeichert sind. Es ist die am weitesten verbreitete Datenbanksprache. Es bietet Möglichkeiten, Beziehungen aufzubauen und Daten zu manipulieren. SQL ist die Standardsprache für relationale Datenbanksysteme. Alle relationalen Datenbankverwaltungssysteme (RDMS), wie MySQL, MS Access, Oracle, Sybase, Informix, Postgres und SQL Server, verwenden SQL als ihre Standard-Datenbanksprache, obwohl sie unterschiedliche „Dialekte“ verwenden:

- MS SQL Server verwendet T-SQL
- Oracle verwendet PL/SQL
- MS Access verwendet eine SQL-Version namens JET SQL (natives Format) usw.

Liste der SQL-Befehle

Es folgt eine Liste von SQL-Befehlen, die alle notwendigen Aktionen mit SQL-Datenbanken abdeckt. Wie bereits erwähnt, kann es jedoch einige Unterschiede zwischen verschiedenen Arten von Datenbanken geben, einschließlich der Verwendung verschiedener „Dialekte“. Jeder SQL-Befehl wird mit seiner Syntax und Beschreibung geliefert.

Die Befehle in SQL werden Abfragen genannt und es gibt zwei Arten:

1. Datendefinitionsabfrage: Die Anweisungen, die die Struktur einer Datenbank definieren, Tabellen erstellen, ihre Schlüssel, Indizes usw.
2. Abfragen zur Datenbearbeitung: Dies sind die Abfragen, die bearbeitet werden können.



2019-1-BG01-KA203-062371

Liste der SQL-Befehle¹:

Befehl	Syntax	Beschreibung
ALTER table	ALTER TABLE table_name ADD column_name datatype;	Es wird verwendet, um einer Tabelle in einer Datenbank Spalten hinzuzufügen
AND	SELECT column_name(s)FROM table_nameWHERE column_1 = value_1 AND column_2 = value_2;	Es ist ein Operator, der verwendet wird, um zwei Bedingungen zu kombinieren
AS	SELECT column_name AS 'Alias'FROM table_name;	Es ist ein Schlüsselwort in SQL, das verwendet wird, um eine Spalte oder Tabelle mit einem Aliasnamen umzubenennen
AVG	SELECT AVG(column_name)FROM table_name;	Es wird verwendet, um eine numerische Spalte zu aggregieren und ihren Durchschnitt zurückzugeben
BETWEEN	SELECT column_name(s)FROM table_nameWHERE column_name BETWEEN value_1 AND value_2;	Es ist ein Operator, mit dem das Ergebnis innerhalb eines bestimmten Bereichs gefiltert wird
CASE	SELECT column_name,CASEWHEN condition THEN 'Result_1'WHEN condition THEN 'Result_2'ELSE 'Result_3'ENDFROM table_name;	Es ist eine Anweisung, die verwendet wird, um verschiedene Ausgaben innerhalb einer SELECT-Anweisung zu erstellen
COUNT	SELECT COUNT(column_name)FROM table_name;	Es ist eine Funktion, die den Namen einer Spalte als Argument nimmt und die Anzahl der Zeilen zählt, wenn die Spalte nicht NULL ist
Create TABLE	CREATE TABLE table_name (column_1 datatype, column_2 datatype, column_3 datatype);	Es wird verwendet, um eine neue Tabelle in einer Datenbank zu erstellen und den Namen der Tabelle und der darin enthaltenen Spalten anzugeben
DELETE	DELETE FROM table_nameWHERE some_column = some_value;	Es wird verwendet, um die Zeilen aus einer Tabelle zu entfernen

¹ Quelle <https://intellipaat.com/blog/tutorial/sql-tutorial/sql-commands-cheat-sheet/>



2019-1-BG01-KA203-062371

GROUP BY	SELECT column_name, COUNT(*)FROM table_nameGROUP BY column_name;	Es ist eine Klausel in SQL, die für Aggregatfunktionen in Zusammenarbeit mit der SELECT-Anweisung verwendet wird
HAVING	SELECT column_name, COUNT(*)FROM table_nameGROUP BY column_nameHAVING COUNT(*) > value;	Es wird in SQL verwendet, da das Schlüsselwort WHERE nicht in Aggregationsfunktionen verwendet werden kann
INNER JOIN	SELECT column_name(s)FROM table_1JOIN table_2 ON table_1.column_name = table_2.column_name;	Es wird verwendet, um Zeilen aus verschiedenen Tabellen zu kombinieren, wenn die JOIN-Bedingung WAHR wird
INSERT	INSERT INTO table_name (column_1, column_2, column_3) VALUES (value_1, 'value_2', value_3);	Es wird verwendet, um einer Tabelle neue Zeilen hinzuzufügen
IS NULL/ IS NOT NULL	SELECT column_name(s)FROM table_nameWHERE column_name IS NULL;	Es ist ein Operator, der mit der WHERE-Klausel verwendet wird, um auf leere Werte zu prüfen
LIKE	SELECT column_name(s)FROM table_nameWHERE column_name LIKE pattern;	Es ist ein spezieller Operator, der mit der WHERE-Klausel verwendet wird, um nach einem bestimmten Muster in einer Spalte zu suchen
LIMIT	SELECT column_name(s)FROM table_nameLIMIT number;	Es ist eine Klausel, um die maximale Anzahl von Zeilen anzugeben, die die Ergebnismenge haben muss
MAX	SELECT MAX(column_name)FROM table_name;	Es ist eine Funktion, die die Anzahl der Spalten als Argument verwendet und den größten Wert davon zurückgibt
MIN	SELECT MIN(column_name)FROM table_name;	Es ist eine Funktion, die die Anzahl der Spalten als Argument verwendet und den kleinsten Wert davon zurückgibt
OR	SELECT column_nameFROM table_nameWHERE column_name = value_1 OR column_name = value_2;	Es ist ein Operator, der verwendet wird, um die Ergebnismenge so zu filtern, dass sie nur die Zeilen enthält, bei denen eine der Bedingungen WAHR ist



2019-1-BG01-KA203-062371

ORDER BY	SELECT column_nameFROM table_nameORDER BY column_name ASC DESC;	Es ist eine Klausel, die verwendet wird, um die Ergebnismenge nach einer bestimmten Spalte entweder numerisch oder alphabetisch zu sortieren
OUTER JOIN	SELECT column_name(s)FROM table_1LEFT JOIN table_2 ON table_1.column_name = table_2.column_name;	Es wird verwendet, um Zeilen aus verschiedenen Tabellen zu kombinieren, auch wenn die Bedingung NICHT WAHR ist
ROUND	SELECT ROUND(column_name, integer)FROM table_name;	Es ist eine Funktion, die den Spaltennamen und eine ganze Zahl als Argument verwendet und die Werte in einer Spalte auf die durch eine ganze Zahl angegebene Anzahl von Dezimalstellen rundet
SELECT	SELECT column_name FROM table_name;	Es ist eine Anweisung, die verwendet wird, um Daten aus einer Datenbank abzurufen
SELECT DISTINCT	SELECT DISTINCT column_nameFROM table_name;	Es wird verwendet, um anzugeben, dass die Anweisung eine Abfrage ist, die eindeutige Werte in angegebenen Spalten zurückgibt
SUM	SELECT SUM(column_name)FROM table_name;	Es ist eine Funktion, die verwendet wird, um die Summe der Werte aus einer bestimmten Spalte zurückzugeben
UPDATE	UPDATE table_nameSET some_column = some_valueWHERE some_column = some_value;	Es wird verwendet, um Zeilen in einer Tabelle zu bearbeiten
WHERE	SELECT column_name(s)FROM table_nameWHERE column_name operator value;	Es ist eine Klausel, die verwendet wird, um die Ergebnismenge zu filtern, um die Zeilen einzuschließen, in denen die WHERE-Bedingung TRUE ist
WITH	WITH temporary_name AS (SELECT *FROM table_name)SELECT *FROM temporary_nameWHERE column_name operator value;	Es wird verwendet, um das Ergebnis einer bestimmten Abfrage in einer temporären Tabelle unter Verwendung eines Alias zu speichern



2019-1-BG01-KA203-062371

Befehle und Syntax zum Abfragen von Daten aus einer einzelnen Tabelle oder mehreren Tabellen²:

Einzelstisch	Mehrere Tische
SELEZIONA c1 DA t Per selezionare i dati della colonna c1 dalla tabella t	SELEZIONA c1, c2 da t1 INNER JOIN t2 su condition Seleziona le colonne c1 e c2 dalla tabella t1 ed esegui un inner join tra t1 e t2
SELEZIONA * DA t Per selezionare tutte le righe e le colonne della tabella t	SELEZIONA c1, c2 da t1 LEFT JOIN t2 su condizione Seleziona la colonna c1 e c2 dalla tabella t1 ed effettua una join sinistra tra t1 e t2
SELEZIONA c1 DA t DOVE c1 = 'test' Per selezionare i dati nella colonna c1 dalla tabella t, dove c1=test	DA t1 RIGHT JOIN t2 su condizione Seleziona la colonna c1 e c2 dalla tabella t1 ed effettua una join a destra tra t1 e t2
SELEZIONA c1 DA t ORDER BY c1 ASC (DESC) Per selezionare i dati nella colonna c1 dalla tabella t in ordine crescente o decrescente	SELEZIONA c1, c2 DA t1 FULL OUTER JOIN t2 su condizione Seleziona la colonna c1 e c2 dalla tabella t1 ed effettua una full outer join tra t1 e t2
SELEZIONA c1 DA t ORDINE PER c1LIMIT n OFFSET offset Per saltare l'offset delle righe e restituire le n righe successive	SELEZIONA c1, c2 DA t1 CROSS JOIN t2 Selezionare la colonna c1 e c2 dalla tabella t1 e produrre un prodotto cartesiano di righe in una tabella
SELEZIONA c1, aggregato(c2) DA t GRUPPO PER c1 Per raggruppare le righe usando una funzione aggregata	SELEZIONA c1, c2 FROM t1, t2 Seleziona la colonna c1 e c2 dalla tabella t1 e produce un prodotto cartesiano di righe in una tabella
SELEZIONA c1, aggregato(c2) DA t GROUP BY c1Condizione HAVING Raggruppare le righe usando una funzione aggregata e filtrare questi gruppi usando la clausola 'HAVING'	SELEZIONA c1, c2 DA t1 A INNER JOIN t2 B su condizione Selezionare la colonna c1 e c2 dalla tabella t1 e unirla a se stessa utilizzando la clausola INNER JOIN

² Quelle <https://intellipaat.com/blog/tutorial/sql-tutorial/sql-commands-cheat-sheet/>



Kommerzielle und kostenlose Datenbanken, die in der realen Welt verwendet werden

• 4th Dimension	• Google Fusion Tables	• MonetDB	• SAND COBMS
• Adabas D	• Greenplum	• mSQL	• SAP HANA
• Airtable	• GroveSite	• MySQL	• SAP Adaptive Server Enterprise
• Apache Derby	• H2	• Netezza	• SAP IQ (formerly known as Sybase IQ)
• Aster Data	• Helix	• NexusDB	• SingleStore
• Amazon Aurora	• HSQLDB	• NonStop SQL	• Snowflake Cloud Data Warehouse
• Allbase	• IBM Business System 12 (historical)	• Nuodb	• solidDB
• CA Datacom	• IBM DB2	• Omnis Studio	• SQL Anywhere (formerly known as Sybase Adaptive Server Anywhere and Watcom SQL)
• CA IDMS	• IBM DB2 Express-C	• OpenLink Virtuoso (Open Source Edition)	• SQLBase
• Clarion	• IBM Lotus Approach	• OpenLink Virtuoso Universal Server	• SQLite
• ClickHouse	• IBM DB2 Express-C	• OpenOffice.org Base	• SQRream DB
• Clustrix	• Infobright	• Oracle	• SAP Advantage Database Server (formerly known as Sybase Advantage Database Server)
• CockroachDB	• Infomix	• Oracle Rdb for OpenVMS	• Teradata
• CSQ	• Ingres	• Panorama	• Tibero
• CUBRID	• InterBase	• Percona	• TIBO
• DataEase	• InterSystems Caché	• Percona Server for MySQL	• TimesTen
• DataFlex	• InterSystems IRIS Data Platform	• Percona XtraDB Cluster	• Trafodion
• Database Management Library	• LibreOffice Base	• Pervasive PSQL	• Unisys RDMS 2200
• Dataphor	• Linter	• Polyhedra	• UniData
• dBase	• MariaDB	• PostgreSQL	• UniVerse
• Derby (aka Java DB)	• MaxDB	• Progress Plus Advanced Server	• Vectorwise
• Empress Embedded Database	• Microsoft Access	• Progress Software	• Vertica
• Exasol	• Microsoft Jet Database Engine (part of Microsoft Access)	• Raima Database Manager (RDM)	• VoltDB
• EnterpriseDB	• Microsoft SQL Server	• R-Base	• YugabyteDB
• eXtremeDB	• Microsoft SQL Server Express	• RethinkDB	
• FileMaker Pro	• SQL Azure (Cloud SQL Server)		
• Firebird	• Microsoft Visual FoxPro		
• FrontBase	• Mimer SQL		

Abbildung 1. Nicht erschöpfende Liste der verfügbaren Datenbanken

Dieser Teil befasst sich mit den gängigen Datenbanken, die auf dem Markt zu finden sind, unabhängig davon, ob sie frei oder proprietär sind. Es stehen jedoch so viele Datenbanken zur Verfügung (Abbildung 1), dass wir nicht alle nennen können. Es musste eine Auswahl getroffen werden und die unten aufgeführten sind die „beliebtesten“ oder die „am häufigsten verwendeten“.

KOMMERZIELLE DATENBANKEN

Aus der Vielzahl der auf dem Markt verfügbaren Datenbanken haben wir drei kommerzielle Datenbanken ausgewählt, die häufig von den großen Unternehmen und Organisationen verwendet werden.



2019-1-BG01-KA203-062371

SAP HANA

Diese Datenbank wird von dem in Deutschland gegründeten europäischen Unternehmen SAP SE entwickelt. SAP HANA ist eine spaltenorientierte Datenbank-Engine, die SAP- und Nicht-SAP-Daten verarbeiten kann. Die Engine wurde entwickelt, um Daten aus Anwendungen und anderen Quellen über mehrere Speicherebenen hinweg zu speichern und abzurufen. SAP HANA kann lokal oder in der Cloud von einer Reihe von Cloud-Service-Providern bereitgestellt werden. Diese Datenbank wird normalerweise von Organisationen gewählt, die Daten aus Anwendungen abrufen und nicht über ein stark eingeschränktes Budget verfügen.



Seine Hauptmerkmale sind:

- Es unterstützt SQL, OLTP und OLAP.
- Die Engine reduziert den Ressourcenbedarf durch Komprimierung.
- Die Daten werden im Speicher abgelegt, wodurch die Zugriffszeiten in einigen Fällen erheblich verkürzt werden.
- Echtzeit-Reporting und Bestandsverwaltung sind verfügbar.
- Es kann mit einer Reihe anderer Anwendungen verbunden werden.

A ab Januar 2021 die aktuell unterstützten Hardware-Plattformen³ für SAP HANA sind:

- Intel-basierte Hardwareplattformen
- IBM Power Systems

A ab Januar 2021 die aktuell unterstützten Betriebssysteme für SAP HANA sind⁴:

- Linux SUSE
- Linux Red Hat

IBM Db2-Datenbank

Die Wurzeln der IBM Db2-Datenbank reichen bis in die frühen 1970er Jahre zurück, als Edgar F. Codd, ein Forscher des Unternehmens, die Theorie der relationalen Datenbanken beschrieb und im Juni 1970 das Modell zur Datenmanipulation veröffentlichte. Heute ist es eine Datenbank-Engine, die über NoSQL-Funktionen verfügt und JSON lesen kann⁵ und XML-Dateien⁶.



³ Quelle SAP SE <https://help.sap.com/viewer/eb3777d5495d46c5b2fa773206bbfb46/2.0.01/en-US/d3d1cf20bb5710149b57fd794c827a4e.html>

⁴ Weitere Informationen zu unterstützten Betriebssystemen für SAP HANA finden Sie im SAP-Hinweis 2235581 - SAP HANA: <https://service.sap.com/sap/support/notes/2235581>

⁵ JavaScript Object Notation ist ein offenes Standarddateiformat als XML und wird als unstrukturierte Daten betrachtet

⁶ XML ist ein offenes Standarddateiformat als JSON und gilt als unstrukturierte Daten



2019-1-BG01-KA203-062371

Die aktuelle Version von DB2 ist LUW 11.1, die eine Vielzahl von Verbesserungen bietet. Eine davon war insbesondere eine Verbesserung der BLU-Beschleunigung (BLink Ultra oder Big Data, Lightning fast und Ultra-easy), die diese Datenbank-Engine durch Data-Skipping-Technologie schneller arbeiten lassen soll. Das Überspringen von Daten wurde entwickelt, um die Geschwindigkeit von Systemen mit mehr Daten zu verbessern, als in den Speicher passen. Die neueste Version von Db2 bietet außerdem verbesserte Disaster-Recovery-Funktionen, Kompatibilität und Analysen.

Seine Hauptmerkmale sind:

- BLU Acceleration kann die verfügbaren Ressourcen für riesige Datenbanken optimal nutzen.
- Es kann aus der Cloud, einem physischen Server oder beidem gleichzeitig gehostet werden.
- Mit dem Taskplaner können mehrere Jobs gleichzeitig ausgeführt werden.
- Fehlercodes und Exitcodes können bestimmen, welche Jobs über den Taskplaner ausgeführt werden.

Die derzeit unterstützten Hardwareplattformen⁷ ab Januar 2021 für IBM Db2 sind:

- IBM z/Architecture-Mainframe
- Intel-basierte Hardwareplattformen

Die derzeit unterstützten Betriebssysteme ab Januar 2021 für IBM Db2 sind:

- z/OS
- Unix
- Linux
- Windows

Oracle-Datenbank

Oracle Database wird häufig für die Ausführung von Online-Transaktionsverarbeitung (OLTP) oder Data Warehousing (DW) verwendet. Es **ORACLE** kann auch OLTP- und DW-Datenbank-Workloads mischen. Oracle Database ist lokal, in der Cloud oder als Hybrid-Cloud-Installation verfügbar. Es kann auf Servern von Drittanbietern sowie auf Oracle Exadata-Hardware vor Ort, in der Oracle Cloud oder in einer privaten Cloud beim Kunden ausgeführt werden.

Die erste Version wurde 1979 veröffentlicht und ihre Entwicklung wurde durch die Forschungen von Edgar F. Codd zum relationalen Datenbankdesign beeinflusst.

Seine Hauptmerkmale sind:

⁷ Quelle IBM Support <https://www.ibm.com/support/pages/system-requirements-ibm-db2-linux-unix-and-windows#1155S>



2019-1-BG01-KA203-062371

- Es ist eine plattformübergreifende Datenbank. Es kann auf verschiedenen Hardware-Betriebssystemen ausgeführt werden, einschließlich Windows Server, Unix und verschiedenen Distributionen von GNU/Linux.
- Es hat seinen Netzwerk-Stack, der es Anwendungen von einer anderen Plattform ermöglicht, reibungslos mit der Oracle-Datenbank zu kommunizieren, zB Anwendungen, die unter Windows laufen, können sich mit der Oracle-Datenbank verbinden, die unter Unix läuft.
- Es handelt sich um eine ACID-konforme Datenbank, die zur Aufrechterhaltung der Datenintegrität und -zuverlässigkeit beiträgt.

Die derzeit unterstützten Hardwareplattformen sind:

- Proprietäre Oracle Database Appliance
- Sparc
- IBM Power Systems
- X64-basierte Hardwareplattformen

Die aktuell unterstützten Betriebssysteme⁸ sind:

- Unix
- Linux
- Windows

KOSTENLOSE DATENBANKEN⁹

Wenn eine Datenbank kostenlos ist, bedeutet dies nicht unbedingt, dass dem Benutzer keine Gebühren in Rechnung gestellt werden. Dies gilt für einige der folgenden Datenbanken, jedoch entscheiden sich einige Entwickler dafür, bestimmte Funktionen einzuschränken und eine Gebühr zu erheben, um diese Funktionen freischalten zu können (siehe die erste Einheit der Grundstufe).

MySQL

MySQL ist eine relationale Open-Source-Datenbank, die auf verschiedenen Plattformen läuft, darunter Windows, Linux, macOS usw. Eine Cloud-Version. MySQL kann für Softwarepakete, geschäftskritische Systeme und Websites mit hohem Volumen verwendet werden.



Seine Hauptmerkmale sind:

⁸ Quelle https://support.oracle.com/knowledge/Oracle%20Database%20Products/1369107_1.html

⁹ Quelle <https://www.guru99.com/free-database-software.html> aktualisiert am 2021



2019-1-BG01-KA203-062371

- Es bietet Skalierbarkeit und Flexibilität
- Das Tool hat Web- und Data Warehouse-Stärken
- Es bietet eine hohe Leistung
- Es verfügt über eine robuste Transaktionsunterstützung

PostgreSQL

PostgreSQL ist ein Open-Source-Datenbankverwaltungssystem der Enterprise-Klasse. Es unterstützt sowohl SQL für relationale als auch JSON für nicht-relationale Abfragen. Es wird von einer erfahrenen Community von Entwicklern unterstützt, die einen enormen Beitrag geleistet haben, um es zu einer äußerst zuverlässigen Datenbankverwaltungssoftware zu machen. Es läuft auf drei verschiedenen Plattformen, nämlich Windows, Linux und macOS. Eine Cloud-Version ist nicht verfügbar. PostgreSQL ermöglicht die Erstellung benutzerdefinierter Datentypen und einer Reihe von Abfragemethoden. Eine gespeicherte Prozedur kann in verschiedenen Programmiersprachen ausgeführt werden.



Seine Hauptmerkmale sind:

- Es ist mit verschiedenen Plattformen kompatibel, die alle gängigen Sprachen und Middleware verwenden
- Standby-Server und Hochverfügbarkeit
- Das Tool verfügt über ausgereifte serverseitige Programmierfunktionen
- Logbasierte und triggerbasierte Replikation SSL
- Es bietet einen hochentwickelten Verriegelungsmechanismus
- Unterstützung für die Kontrolle der Parallelität mehrerer Versionen
- Es bietet Unterstützung für die Client-Server-Netzwerkarchitektur
- Das Tool ist objektorientiert und ANSI-SQL2008 kompatibel
- PostgreSQL ermöglicht die Verknüpfung mit anderen Datenspeichern wie NoSQL, die als föderierter Hub für mehrsprachige Datenbanken fungieren.

Microsoft SQL

SQL Server ist ein von Microsoft entwickeltes RDBMS. Es unterstützt ANSI SQL, die Standardsprache von SQL (Structured Query Language). SQL Server wird jedoch mit seiner Implementierung der SQL-Sprache T-SQL (Transact-SQL) geliefert. Es läuft auf Docker Engine, Ubuntu, SUSE Linux Enterprise Server und Red Hat Enterprise Linux. Eine Cloud-Version ist verfügbar.





2019-1-BG01-KA203-062371

Seine Hauptmerkmale sind:

- Es bietet die Integration strukturierter und unstrukturierter Daten mit der Leistungsfähigkeit von SQL Server und Spark.
- Das Tool bietet Skalierbarkeit, Leistung und Verfügbarkeit für geschäftskritische, intelligente Anwendungen, Data Warehouses und Data Lakes.
- Es bietet erweiterte Sicherheitsfunktionen zum Schutz Ihrer Daten.
- Zugriff auf umfangreiche, interaktive Power BI-Berichte, um schnellere und bessere Entscheidungen zu treffen.

MariaDB

MariaDB ist ein Fork des MySQL-Datenbankverwaltungssystems. Es wurde von seinen ursprünglichen Entwicklern erstellt. Dieses DBMS-Tool bietet Datenverarbeitungsfunktionen für kleine und große Unternehmen. Es läuft auf drei Plattformen, nämlich Windows, Linux und macOS. Eine Cloud-Version ist verfügbar. MariaDB ist eine alternative Software zu MySQL. Es bietet eine hohe Skalierbarkeit durch einfache Integration.



Seine Hauptmerkmale sind:

- Es arbeitet unter GPL-, BSD- oder LGPL-Lizenzen.
- Es wird mit vielen Speicher-Engines geliefert, einschließlich der Hochleistungs-Engines, die in andere relationale Datenbankverwaltungssysteme integriert werden können.
- Es bietet die Galera-Cluster-Technologie.
- MariaDB kann auf verschiedenen Betriebssystemen laufen und unterstützt zahlreiche Programmiersprachen.

Orakel

Oracle ist eine sich selbst reparierende, selbstsichernde und selbstfahrende Datenbank, die entwickelt wurde, um die manuelle Datenverwaltung zu eliminieren. Es ist eine intelligente, sichere und hochverfügbare Datenbank in der Cloud, die Unternehmen beim Wachstum unterstützt. Es läuft auf zwei Plattformen, nämlich Windows und Linux. Eine Cloud-Version ist ebenfalls verfügbar.



Seine Hauptmerkmale sind:

- Oracle Cloud ist für leistungsstarke Datenbank-Workloads, Streaming-Workloads und Hyperscale-Big Data optimiert.
- Sie können problemlos in die Cloud migrieren.



2019-1-BG01-KA203-062371

- Es stellt die Dienste basierend auf Ihrer Betriebsweise bereit, um Oracle Cloud in Ihrem Rechenzentrum auszuführen.

Firebirdsql

Firebird ist ein Open-Source-SQL-RDBMS, das auf Microsoft Windows, macOS, Linux und mehreren Unix-Plattformen, einschließlich HP-UX, Solaris und AIX, läuft. Eine Cloud-Version ist verfügbar. Firebird bietet entwicklungsfreundliche Sprachunterstützung, gespeicherte Prozeduren und Trigger.



Seine Hauptmerkmale sind:

- Firebird ermöglicht es Ihnen, eine benutzerdefinierte Version zu erstellen.
- Es ist kostenlos herunterzuladen, zu registrieren und bereitzustellen.
- Das Tool verfügt über ein verbessertes Multi-Plattform-RDBMS.
- Bietet eine Reihe von Finanzierungsmöglichkeiten von firebird-Mitgliedschaften bis hin zu Sponsoring-Verpflichtungen.

Datenbanken in der wissenschaftlichen Welt Aufbaumodul

Dieser Abschnitt widmet sich der weiteren Erforschung der in der Wissenschaft verwendeten Open-Access-Datenbanken und der Nutzung und Nutzung des vorhandenen Wissens.

ÜBERSICHT ÜBER DATENBANKEN IN DER WISSENSCHAFTLICHEN WELT

Bestehende Datenbanken, die der Wissenschaft gewidmet sind und wie man sie verwendet

Wie bereits erwähnt, ist das Teilen, Integrieren und Kommentieren von Daten ein wesentlicher Bestandteil der biologischen Forschung, da es den Forschern ermöglicht, die Untersuchung und Interpretation experimenteller Ergebnisse zu reproduzieren. Obwohl angenommen wird, dass Bioinformatiker und Informatiker für diese Aktionen verantwortlich sind, spielen die Biowissenschaftler eine gleichberechtigte Rolle bei der Förderung der Datenintegration, da sie diejenigen sind, die diese Art von Daten generieren und in der Regel die Endnutzer sind.



2019-1-BG01-KA203-062371

Datenintegration ist definiert als der Prozess der Kombination von Daten aus verschiedenen Quellen, um Benutzern eine einheitliche Sicht auf diese Daten zu bieten. In den Computerwissenschaften wurden die theoretischen Rahmenbedingungen für die Datenintegration basierend auf der Methode zur Datenintegration in „eager“ und „faul“ kategorisiert. Nach der Eager-Methode, auch Warehousing genannt, werden die Daten in ein globales Schema kopiert und in einem zentralen Data Warehouse gespeichert. Der Begriff „Schema“ bezieht sich auf einen organisierten und „abfragbaren“ Ansatz zum Speichern von Daten. Bei der Lazy-Methode befinden sich die Daten in verteilten Quellen und werden bei Bedarf gemäß einem globalen Schema integriert, das zum Mapping der Daten zwischen den Quellen verwendet wird. Das Datenvolumen, der Eigentümer der Daten und die vorhandene Infrastruktur sind die Hauptfaktoren, die letztendlich bestimmen, welche der beiden Methoden für die Datenintegration verwendet wird. Darüber hinaus können diese Methoden in den biologischen Wissenschaften auf verschiedene Weise und auf verschiedenen Ebenen angewendet werden. Als Ergebnis wurden sechs verschiedene und weit verbreitete Schemata für die Integration von Daten formuliert:

- Datenzentralisierung: Die Daten befinden sich in zentralisierten Ressourcen. UniProt und GenBank sind zwei Beispiele für Datenbanken, die dieser Methode folgen.
- Data Warehousing: Daten aus verschiedenen Quellen befinden sich in einem zentralen Repository. Pathway Commons ist eine Datenbank, die diesem Ansatz folgt, um Daten zu integrieren.
- Dataset-Integration: Interne Workflows greifen auf verteilte Datenbanken zu und laden Daten in ein lokales Repository herunter.
- Hyperlinks: Dieser Ansatz ermöglicht Benutzern den Zugriff auf Datenbanken und Tools in verschiedenen Bereichen der Biowissenschaften und fördert so die Interoperabilität. ExPASy ist ein indikatives Beispiel für ein Portal, das auf dieser Datenintegrationsmethodik basiert.
- Föderierte Datenbanken: Für die Integration von Daten in heterogene Datenbanken ist eine Translationsschicht erforderlich. Das bedeutet, dass Daten aus der Datenbank so in ein allgemein akzeptiertes Format umgewandelt werden, dass sie von einem Mapping-Dienst in gleicher Weise interpretiert werden können. Das Distributed Annotation System (DAS), ein Client-Server-System, ist ein indikatives Beispiel.
- Linked Data: Ein Netzwerk miteinander verbundener Daten, auf die online zugegriffen werden kann. Grafische Benutzeroberflächen (GUI), die aus Hyperlinks bestehen, die verknüpfte Daten von zahlreichen Datenanbietern verbinden und somit ein großes System von Linked Data bilden. BIO2RDF ist ein indikatives Beispiel für eine Datenbank, die diesen Ansatz als Grundlage für die Datenintegration verwendet.

Datenzentralisierung, Data Warehousing und Datensatzintegration basieren auf dem „eager“ theoretischen Rahmen, während Hyperlinks, föderierte Datenbanken und verknüpfte Daten auf dem „faulen“ theoretischen Rahmen bezüglich der Art und Weise, die für die Datenintegration ausgewählt wird, basieren.



2019-1-BG01-KA203-062371

Datenformate werden als organisierte Möglichkeit zur Demonstration von Daten und Metadaten in einer Datei beschrieben. Wissenschaftler begannen, biologische Daten in formatierten Dateien zu speichern, da das exponentielle Wachstum der Daten die Notwendigkeit schuf, sie mithilfe von Computersystemen und Datenbanken zu analysieren. Ein Problem, das bei der Dateiformatierung aufgetreten ist, ist das Aufkommen verschiedener Formate, selbst für die Darstellung der gleichen Art von Daten. In einigen Fällen wurde beobachtet, dass mehr als eine Formatklasse verwendet werden kann, um die Daten und Metadaten in einer einzigen Datei darzustellen. Darüber hinaus hat die Forschung gezeigt, dass die am häufigsten verwendeten Formatklassen sind: i) Tabellen, ii) FASTA-ähnlich, iii) Tag-strukturiert und iv) GenBank-ähnlich. Die ideale Lösung für dieses Problem wäre, dass sich Wissenschaftler auf die Verwendung einer begrenzten Anzahl spezifischer Formate einigen, um den Prozess der Datenintegration zu vereinfachen. Auch das Design von Konvertern, die alle unterschiedlichen Formatklassen übersetzen können, wäre eine hilfreiche Lösung.

Derzeit werden über 1.700 Datenbanken mit Daten von biologischem Interesse verwendet, so die nicht erschöpfende Liste, die von der Zeitschrift *Nucleic Acids Research* kuratiert wurde. Um für einen bestimmten Zweck als wertvoll erachtet zu werden, müssen alle Datensätze, die in einer Datenbank vorhanden sind, integriert und strukturiert werden. Die bestehenden biologischen Datenbanken umfassen Informationen zu einer Vielzahl von biologischen Forschungsthemen.

Wie bereits im Basic Level erwähnt, hängt die Klassifizierung biologischer Datenbanken von mehreren Faktoren ab, darunter der Umfang der Datenabdeckung und der Grad der Biokuration. Dennoch ist ihre Klassifikation nach Art der Daten eine der einfachsten und umfassendsten Möglichkeiten, biologische Datenbanken zu kategorisieren. Daher werden diese im folgenden Abschnitt als DNA-, RNA-, Protein-, Krankheits-, Expressions- und Pathway-Datenbanken beschrieben.

DNA-Datenbanken

DNA-Datenbanken konzentrieren sich auf den Umgang mit DNA-Daten von zahlreichen oder wenigen bestimmten Arten. Der Hauptzweck menschlicher DNA-Datenbanken besteht darin, das Referenzgenom zu erstellen, ein Profil der menschlichen genetischen Variation durchzuführen, den Genotyp mit dem Phänotyp zu verbinden und menschliche Mikrobiom-Metagenome zu identifizieren. Ein Beispiel für eine DNA-Datenbank ist GenBank, eine öffentlich zugängliche Sammlung aller untersuchten DNA-Sequenzen. Ab Februar 2021 sind in der GenBank (<http://www.ncbi.nlm.nih.gov/genbank/statistics>) über 776 Milliarden Nukleotidbasen in über 226 Millionen Sequenzen verfügbar.

RNA-Datenbanken

Diese Datenbanken enthalten Informationen über nicht-kodierende RNAs (ncRNAs), wie microRNAs und lange nicht-kodierende RNAs (lncRNAs), die keine Proteine kodieren. Der Zweck von



2019-1-BG01-KA203-062371

RNA-Datenbanken besteht darin, ncRNAs, von denen lncRNAs am häufigsten untersucht werden, zu entschlüsseln und ihre Funktionen und Wechselwirkungen zu beschreiben. Ein Beispiel für eine RNA-Datenbank ist RNACentral, die aus einer einheitlichen Ansicht von ncRNA-Sequenzdaten besteht, die aus einer Reihe von Datenbanken stammen, von denen einige Rfam, miRBase und lncRNAdb sind.

Proteindatenbanken

Proteindatenbanken wurden entwickelt, um eine umfangreiche Zusammenstellung universeller Proteine zu erstellen, Proteinfamilien und -domänen zu identifizieren, phylogenetische Bäume zu rekonstruieren und Proteinstrukturen zu profilieren. PDB, das aus Tausenden von Strukturen biologischer Makromoleküle besteht, ist ein indikatives Beispiel für Proteindatenbanken.

Krankheitsdatenbanken

Krankheitsdatenbanken enthalten definitionsgemäß Informationen über verschiedene Arten von Krankheiten, konzentrieren sich jedoch hauptsächlich auf die Bereitstellung von Daten zu verschiedenen Krebsarten. Eines der wichtigsten Krebsprojekte, das entwickelt wurde, ist der Cancer Genome Atlas (TCGA), dessen Ziel es ist, ein breites Spektrum an Omics-Daten wie mRNA, SNP und Methylierung für über zwanzig verschiedene Krebsformen beim Menschen zu sammeln.

Ausdrucksdatenbanken

Expressionsdatenbanken können für eine Reihe von Aufgaben verwendet werden, wie zum Beispiel das Studium der gewebespezifischen Genexpression und -regulation, das Speichern von Expressionsdaten, das Detektieren der differentiellen und Grundlinienexpression und das Untersuchen und Überprüfen von Expressionsinformationen, die aus RNA- und Proteindaten erhalten wurden. Als Expressionsdatenbank enthält der Human Protein Atlas Expressionsprofile für einen signifikanten Prozentsatz der humanen proteinkodierenden Gene, die aus RNA- und Proteindaten abgeleitet wurden.

Pfaddatenbanken

Pathway-Datenbanken enthalten Daten über biologische Wege, die von Forschern zur Analyse von Stoffwechsel-, Regulations- und Signalwegen genutzt werden können. Ein charakteristisches Beispiel für Pathway-Datenbanken ist KEGG PATHWAY, das Informationen über molekulare Interaktionen und Reaktionsnetzwerke enthält.

Das National Center for Biotechnology Information (NCBI), Teil der US-amerikanischen National Library of Medicine am National Institute of Health, hat ein integriertes Datenbankabrufsystem entwickelt, das Zugriff auf 34 verschiedene Datenbanken mit insgesamt 3,0 Milliarden Datensätzen namens Entrez bietet. Die globale Suchseite von Entrez (<https://www.ncbi.nlm.nih.gov/search/>) bietet für jede der 34 Datenbanken Links zum Webportal. Das Entrez-System ist einfach zu bedienen, da es Benutzern ermöglicht, Daten in einer Vielzahl von Formaten herunterzuladen und mithilfe einfacher



2019-1-BG01-KA203-062371

boolescher Abfragen eine Textsuche durchzuführen. Datensätze werden zwischen Datenbanken auf der Grundlage von behaupteten Beziehungen verknüpft; diese Datensätze können in verschiedenen Formaten dargestellt werden. Darüber hinaus haben Benutzer von Entrez die Möglichkeit, einzelne Datensätze oder Datensatzstapel herunterzuladen. Einige der 34 Datenbanken, die Teil von Entrez sind, sind die folgenden: PubMed (<https://pubmed.ncbi.nlm.nih.gov>), das wissenschaftliche und medizinische Zusammenfassungen/Zitate enthält; BioSample (<https://www.ncbi.nlm.nih.gov/biosample>), das Beschreibungen von biologischen Ausgangsmaterialien umfasst; GEO Profiles (<https://www.ncbi.nlm.nih.gov/geoprofiles>), die Genexpressions- und molekulare Häufigkeitsprofile umfassen; und, dbVar (<https://www.ncbi.nlm.nih.gov/dbvar>).

Die an das NCBI übermittelten Daten stammen aus drei Quellen: i) direkt von Forschern, ii) nationalen und internationalen Partnerschaften oder Vereinbarungen mit Datenlieferanten und Forschungskonsortien und iii) internen Kurationsbemühungen. Bemerkenswert ist, dass das NCBI für die Verwaltung der GenBank-Datenbank verantwortlich ist und an der International Nucleotide Sequence Database Collaboration (INSDC) in Zusammenarbeit mit dem EMBL-EBI European Nucleotide Archive (ENA) und der DNA Data Bank of Japan (DDBJ) beteiligt ist.

Da sich Datenbanken in vielen wissenschaftlichen Bereichen als nützliches Werkzeug erwiesen haben, gewinnt ihr Einsatz im Gesundheitswesen stetig an Bedeutung. Heutzutage haben technologische Fortschritte im Bereich der Datenwissenschaft es medizinischen Fachkräften ermöglicht, gesundheitsbezogene Daten zu sammeln, zu verarbeiten und zu analysieren, was nicht nur zu einer Verbesserung der Versorgung, sondern auch der Sicherheit von Patienten und Verbrauchern führt. Damit diese Verbesserungen stattfinden können, müssen relevante Daten effizient und sicher erfasst, gespeichert, analysiert und über die verschiedenen Leistungsstufen eines Gesundheitssystems hinweg ausgetauscht werden. Dies hat zur Entwicklung von elektronischen Gesundheitsakten (EHRs) geführt, Datenbanken, die Patientendaten speichern, auf die medizinisches Fachpersonal zugreifen und sie nutzen kann.

EHRs können als medizinische Datenbanken definiert werden, die Benutzern, in diesem Fall medizinisches Fachpersonal und Verwaltungspersonal, Zugang zu Gesundheitsakten bieten. Die unterschiedlichsten Arten von EHRs sind die elektronische Krankenakte (EMR) und die persönliche Gesundheitsakte (PHR). EMRs bestehen aus Informationen, die von einer einzelnen Krankenhausabteilung, einem ganzen Krankenhaus oder Teilen des Krankenhauses eingereicht werden. Sie können auch Informationen aus einer Reihe von Krankenhäusern enthalten. Informationen zu dieser Art von EHR werden normalerweise nur vom Krankenhauspersonal hinzugefügt. Im Gegenteil, PHRs werden von den Patienten verwaltet, die Informationen eingeben können. PHRs werden als elektronische Anwendungen beschrieben, die Patienten eine sichere Plattform bieten, um ihre Gesundheitsdaten zu kontrollieren und zu teilen. Der Hauptunterschied zwischen den beiden Arten von EHR-Systemen besteht darin, dass in PHRs

Das erste EHR-System wurde in den 1960er Jahren hauptsächlich aufgrund des Aufbaus von unstrukturierten und ungenutzten Patienteninformationen über einen Zeitraum von mehreren Jahrzehnten verfügbar. Große Organisationen begannen, Datenbanksysteme einzurichten, um Daten in



2019-1-BG01-KA203-062371

zentralen Repositories zu speichern und zu strukturieren. Diese Datenbanken ermöglichen die Organisation und Sammlung von Daten aus vielen verschiedenen Quellen, darunter Apotheken, Labors, klinische Studien und Bestandteile der klinischen Versorgung, wie z. B. Aufzeichnungen über die Verabreichung von Medikamenten. Derzeit wird die Implementierung von EHR-Systemen hauptsächlich in Ländern mit hohem Einkommen beobachtet. Beispielsweise führte der Health Information Technology for Economic and Clinical Health Act (HITECH Act von 2009) zur Digitalisierung des Gesundheitsversorgungssystems in den USA und zur anschließenden Entwicklung der Medicare- und Medicaid-EHR-Incentive-Programme.

Der Hauptzweck für die Erstellung von EHRs war die Notwendigkeit, Patientenakten zu archivieren und zu strukturieren. Sie wurden später aus Abrechnungs- und Qualitätsverbesserungsgründen benannt. Mit dem technologischen Fortschritt wurden EHRs im Laufe der Jahre integrativer, dynamischer und vernetzter. Dennoch wird Big Data im Vergleich zu anderen Branchen in der Medizinbranche nicht optimal genutzt. Dies geschah hauptsächlich aufgrund der schlechten Qualität der gesammelten Daten und schlecht strukturierter Datensätze. Vor der Entwicklung von EHRs basierte die medizinische Forschung auf Krankheitsregistern oder chronischen Krankheitsmanagementsystemen (CDMS). Diese Repositorien weisen erhebliche Einschränkungen auf, da sie aus Datensammlungen bestehen, die sich oft nur auf eine bestimmte Krankheit beziehen. Weiter, sie können die Daten oder Schlussfolgerungen nicht auf andere Krankheiten übertragen und können Informationen von einer Patientengruppe in einem bestimmten geografischen Gebiet enthalten. Andererseits sind EHR-Daten sehr vielfältig und erleichtern so die Analyse komplexer klinischer Interaktionen und Entscheidungen.

Die Bestandteile von EHRs sind verschiedene Arten medizinischer Daten, die von Gesundheitsakten bis hin zu sensorischen Rohdaten reichen. Medizinische Daten können in sensible Daten oder nicht sensible Daten kategorisiert werden. Sensible Daten umfassen Patienteninformationen oder können einem Patienten zugeordnet werden. Zu den nicht sensiblen Daten zählen sensorische Daten, die auch Messdaten genannt werden, da sie nur aus Proben von Sensoren bestehen, wie beispielsweise Proben einer EEG-Messung. Daten, die in einer medizinischen Datenbank gespeichert sind, werden als Metadaten bezeichnet. Der am häufigsten verwendete Datenbanktyp zum Speichern medizinischer Daten ist die relationale Datenbank, die Daten in Form von Tabellen präsentiert, die aus Zeilen und einer festgelegten Anzahl von Spalten bestehen. Einige Datenbanken können Patienteninformationen wie die Krankengeschichte eines Patienten oder anonymisierte Daten enthalten, die in Studien verwendet werden können.

Medizinische Daten können wie nachfolgend beschrieben in mehrere Kategorien unterteilt werden:

- **Medizin- und Labordaten:** Medizinisches Personal kann in einem ärztlichen Verordnungserfassungssystem Verordnungen für Medikamente oder Laboruntersuchungen einreichen, die anschließend vom Labor- oder Pflegepersonal durchgeführt werden. Beispiele für diese Datenkategorie sind Arzneimittelverordnungen und mikrobiologische Ergebnisse.



2019-1-BG01-KA203-062371

- Abrechnungsdaten: Diese Kategorie medizinischer Daten umfasst Codes, die von Krankenhäusern verwendet werden, um Ansprüche bei ihren Versicherungsanbietern geltend zu machen. Die von der WHO erstellte Internationale Klassifikation der Krankheiten und die von der American Medical Association unterstützte Current Procedural Terminology sind die beliebtesten Kodierungssysteme.
- Bilder: Dies können Röntgenbilder sein, die aus Röntgenaufnahmen, Echokardiogrammen und Computertomographie (CT)-Scans resultieren.
- Hinweise und Berichte: Diese können mit dem Fortschritt der Patienten in Verbindung gebracht werden. Entlassungszusammenfassungen gehören ebenfalls in diese Kategorie. Befunde aus bildgebenden Untersuchungen werden in der Regel in Operationsberichten beschrieben. Notizen müssen teilweise mit einem Vorlagensystem strukturiert werden.
- Physiologische Daten: Diese Kategorie medizinischer Daten enthält Vitalparameter wie Herzfrequenz und Blutdruck sowie EKG- und EEG-Kurven.

Relationale Datenbanken werden am häufigsten für die Verwaltung und Speicherung medizinischer Daten verwendet. Sie können als eine Sammlung von Tabellen bezeichnet werden, die durch gemeinsame Schlüssel verbunden sind. Ein Datenbankschema bestimmt die Struktur der Tabellen und ihre Beziehungen. Eine einfache medizinische Datenbank kann vier Tabellen enthalten:

- Tabelle 1: eine Patientenliste
- Tabelle 2: ein Krankenhausaufnahmeprotokoll
- Tabelle 3: eine Liste mit Vitalparametermessungen
- Tabelle 4: ein Wörterbuch mit Vitalzeichencodes und zugehörigen Labels

Zur Verknüpfung der vier Tabellen können Primär- und Fremdschlüssel verwendet werden.

Das Überwiegen von Gesundheitsdatenbanken bietet aus verschiedenen Gründen einen eingeschränkten Zugang zu Daten, einschließlich Datenschutzbedenken und Pläne zur Monetarisierung der Daten. Nichtsdestotrotz stehen eine Reihe von Open-Access-Gesundheitsdatenbanken für die öffentliche Nutzung zur Verfügung, von denen einige im Folgenden beschrieben werden.

Die Datenbank des Medical Information Mart for Intensive Care (MIMIC)

Die MIMIC-Datenbank (<http://mimic.physionet.org>) entstand 2003 als Ergebnis einer Zusammenarbeit zwischen dem MIT, Philips Medical Systems und dem Beth Israel Deaconess Medical Center (BIDMC). Die in diese Datenbank eingegebenen Daten stammen von medizinischen und chirurgischen Patienten, die auf allen Intensivstationen des BIDMC aufgenommen wurden. Es besteht aus Informationen von über vierzigtausend Patienten, detaillierten physiologischen und klinischen Daten und ist anonymisiert und für Forscher offen zugänglich. In dieser Datenbank sind zwei Arten von Daten vorhanden: klinische Daten, die von EHRs abgeleitet werden, die in einer relationalen Datenbank mit



2019-1-BG01-KA203-062371

ungefähr 50 Tabellen gespeichert sind, und Wellenformen des Bettmonitors, die in flachen Binärdateien gespeichert sind.

PCORnet

PCORnet, das National Patient-Centered Clinical Research Network, ist eine Initiative, die 2013 mit dem Ziel begann, Daten aus mehreren Clinical Data Research Networks und Patient-Powered Research Networks zu integrieren. Es enthält 29 Netzwerke, die den Zugang zu umfangreichen Forschungsergebnissen erleichtern. Es sammelt Daten von routinemäßigen Patientenbesuchen und Daten, die von einzelnen Patienten über persönliche Gesundheitsakten oder Community-Netzwerke mit anderen Patienten geteilt werden.

NHS öffnen

Der National Health Services (NHS England) unterhält eines der größten Datenarchive der Welt mit Daten zur Gesundheit der Bevölkerung. NHS öffnen¹⁰ ist eine Open-Source-Datenbank, die Zugang zu Informationen bietet, die der Öffentlichkeit von der Regierung oder anderen öffentlichen Stellen zur Verfügung gestellt werden. Dieses Projekt wurde ins Leben gerufen, um die Transparenz zu erhöhen und die Effizienz des britischen Gesundheitssektors zu überwachen. Patienten, Beschäftigte im Gesundheitswesen und Beauftragte erhalten die Möglichkeit, die Versorgungsqualität an verschiedenen Orten des Landes zu vergleichen, indem sie einfach auf die verfügbaren Daten in der speziell dafür eingerichteten Datenbank zugreifen.

De-Identifikation der Datenbank

Einer der wichtigsten Schritte zum Aufbau einer EHR-Datenbank ist die Anonymisierung. Bevor eine Datenbank für Forscher und Anwendungen zur Verfügung steht, müssen unbedingt Maßnahmen ergriffen werden, um sicherzustellen, dass Datenschutzrichtlinien und -vorschriften eingehalten werden. Bei strukturierten Daten wie Spalten einer Tabelle basiert die De-Identifikation auf der Kategorisierung von Daten und der anschließenden Löschung oder Kryptographie der als geschützt gekennzeichneten Daten. Für unstrukturierte Daten, wie zum Beispiel Entlassungszusammenfassungen, werden verschiedene Techniken der natürlichen Sprachverarbeitung verwendet, von einfachen regulären Ausdrücken bis hin zu komplexen neuronalen Netzen, die versuchen, alle durch Freitext geschützten Informationen zu finden, um eine Löschung oder Kryptographie durchzuführen.

¹⁰ Offene Daten beim NHS. Verfügbar unter: <http://www.england.nhs.uk/ourwork/tsd/data-info/open-data/>



2019-1-BG01-KA203-062371

DIE ANWENDUNG VON BLOCKCHAIN IN DER DIGITALEN GESUNDHEIT

Die Blockchain-Technologie basiert auf dem Konzept eines dezentralen Systems zur Datenspeicherung, bei dem jedem Teilnehmer/Knoten eine Kopie des Ledgers der durchgeführten Transaktionen zur Verfügung gestellt wird. Dies macht es für jemanden unmöglich, die Daten zu ändern, ohne dass die anderen Teilnehmer informiert werden. Starke zentralisierte Einheiten würden von der Anwendung der Blockchain profitieren. Die Anwendungen von Digital Health hängen stark von zentralisierten Systemen ab. Daher hat die Blockchain das Potenzial, die digitale Gesundheit zu verändern, indem sie die Art und Weise ändert, wie Daten gespeichert und gesichert werden. Für seine Anwendung wurden verschiedene Bereiche vorgeschlagen, darunter Lieferketten, Arzneimittelverifizierung, Erstattung von Ansprüchen, Zugangskontrolle und klinische Studien.

Medizinische Daten haben sich als die am höchsten bewerteten Daten von Hackern erwiesen, da neuere Studien geschätzt haben, dass eine einzelne Gesundheitsakte bis zu 400 USD kosten kann. Daher ist die sichere Aufbewahrung der Daten in medizinischen Datenbanken von größter Bedeutung. Blockchain kann eine Lösung für dieses Problem bieten, indem Datenschutz, Integrität, Authentifizierung und Autorisierung sichergestellt werden. Blockchain-Daten werden verschlüsselt, und wenn jemand seine Daten löschen oder unbrauchbar machen muss, erhält er diese Möglichkeit, indem er einen Schlüsselzerstörungsmechanismus anwendet, bei dem der Schlüssel, der ursprünglich für die Verschlüsselung der Nachricht verwendet wurde, zerstört oder unbrauchbar gemacht wird. Danach sind die in der Blockchain gespeicherten Daten nicht lesbar.

Blockchain ist in der Lage, zwei wesentliche Bedürfnisse in Bezug auf den Datenaustausch zu erfüllen: Integrität und Nichtabstreitbarkeit. Integrität bedeutet, dass die Abfrage und die abgerufenen Daten nicht mehr geändert werden können, nachdem der Abrufvorgang ausgeführt wurde. Nichtabstreitbarkeit bedeutet, dass der Wissensabrufdienst nicht die Fähigkeit besitzt, zu leugnen, dass die spezifischen Daten von dem Dienst als Antwort auf eine bestimmte Anfrage zu einem bestimmten Zeitpunkt geliefert wurden. Blockchain kann als ein verteiltes Transaktionsmanagementsystem definiert werden, das nicht beschädigt werden kann. Es kann für die EHR-Integration, gemeinsame Nutzung und Zugriffskontrolle, Aufbewahrung und Verwaltung verwendet werden.

Ein theoretischer blockchainbasierter Notardienst kann aus drei Rechenschichten bestehen:

- ein Datenkonsumenten-Front-End
- eine Schnittstelle zur Kommunikation mit biomedizinischen Datenbankschnittstellen, und
- die Vertrags-Engine, die die Abfrage organisiert und die abgerufenen Ergebnisse an den Verbraucher zurückgibt, Transaktionen durchführt und vorbereitet und Verträge und deren Metadaten verwaltet

Für die Anwendung des Notardienstes können zwei verschiedene Schemata verwendet werden: das Basisschema und das Versionierungsschema. Das grundlegende Schema wendet ein Abfrage-Antwort-Ledger an, durch das der Benutzer einen versiegelten Nachweis erhält, der bestätigt, dass zu



2019-1-BG01-KA203-062371

einem bestimmten Zeitpunkt eine bestimmte Abfrage in einer biomedizinischen Datenbank platziert wurde, die bestimmte Ergebnisse liefert. Dieses Schema kann verwendet werden, um die Integrität und Nichtabstreitbarkeit einer Anfrage sicherzustellen, wenn eine lebenswichtige biomedizinische Aufgabe auf der spezifischen Anfrage beruht. Das Versionierungsschema ermöglicht die nicht seriöse Versionierung von Daten, die zu zahlreichen Gelegenheiten aus einer sich dynamisch entwickelnden biomedizinischen Datenbank abgerufen wurden, wobei immer dieselbe Abfrage verwendet wird. Dieses Schema kann angewendet werden, um verschiedene Versionen von sich ändernden medizinischen Nachweisen zu bestätigen, wie sie aus einer biomedizinischen Datenbank mit häufig aktualisiertem Inhalt abgerufen werden.

Die Integration der Blockchain-Technologie in pharmazeutische oder biowissenschaftliche Anwendungen hat die Fähigkeit, die Schnittstelle und den Datenaustausch zu dezentralisieren, was zu mehr Effizienz, höheren Geschwindigkeiten und unbegrenzter Skalierbarkeit führt. Blockchain macht Daten unveränderlich, was in klinischen Studien nützlich wäre, um sicherzustellen, dass klinische Daten zu einem späteren Zeitpunkt nicht von Forschern manipuliert werden können. Es kann auch bei der Identifizierung, Rückverfolgung und Verifizierung von Arzneimitteln verwendet werden. Mit der Implementierung der Blockchain sind bestimmte Risiken verbunden, wie Datenschutzbedenken, Transaktionen außerhalb der Kette und Zweifel an dieser Technologie aufgrund mangelnder Akzeptanz. Nichtsdestotrotz überwiegen die Vorteile der Blockchain-Technologie die möglichen Nachteile bei weitem und könnten eine bedeutende Rolle bei der Begrenzung der Methoden spielen, die für illegale Aktivitäten verwendet werden.



2019-1-BG01-KA203-062371

Verweise

Agha-Mir-Salim L, Sarmiento RF. 2020. Health information technology as premise for data science in global health: A discussion of opportunities and challenges. In: *Leveraging Data Science for Global Health*. Cham: Springer International Publishing, 3–15.

Amid C, Alako BTF, Balavenkataraman Kadhivelu V, Burdett T, Burgin J, Fan J, Harrison PW, Holt S, Hussein A, Ivanov E et al. 2020. The European nucleotide archive in 2019. *Nucleic Acids Res.*, 48:D70–76.

Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, et al. 2004. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res.*, 32 (Suppl 1):115–9. doi: 10.1093/nar/gkh131.

Artimo P, Jonnalagedda M, Arnold K, Baratin D, Csardi G, de Castro E, et al. 2012. ExPASy: SIB bioinformatics resource portal. *Nucleic Acids Res.*, 40(Web Server issue):597–603. doi: 10.1093/nar/gks400.

Belleau F, Nolin MA, Tourigny N, Rigault P, Morissette J. 2008. Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *J Biomed Inform.*, 41(5):706–16.

Benson DA, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. 2014. GenBank. *Nucleic Acids Res.*, 42:D32–D37.

Benson DA, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. 2014. GenBank. *Nucleic Acids Res.*, 42:D32–D37.

Benson DA, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. 2014. GenBank. *Nucleic Acids Res.*, 42:D32–D37.

Bornberg-Bauer E, Paton NW. 2002. Conceptual data modelling for bioinformatics. *Brief Bioinform.*, 3(2):166–80.

Bulgarelli L, Núñez-Reiz A, Deliberato RO. 2020. Building electronic health record databases for research. In: *Leveraging Data Science for Global Health*. Cham: Springer International Publishing, 55–64.

Burge SW, Daub J, Eberhardt R, Tate J, Barquist L, Nawrocki EP, et al. 2013. Rfam 11.0: 10 years of RNA families. *Nucleic Acids Res.*, 41: D226-232.

Cancer Genome Atlas Research Network, Weinstein JN, Collisson EA, Mills GB, Shaw KR, Ozenberger BA, et al. 2013. The Cancer Genome Atlas Pan-Cancer analysis project. *Nat Genet.*, 45: 1113-1120.

Cerami EG, Gross BE, Demir E, Rodchenkov I, Babur O, Anwar N, et al. 2011. Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Res.*; 39(Database issue): 685–90.



2019-1-BG01-KA203-062371

Chavali LN, Prashanti NL, Sujatha K, Rajasheker G, Kavi Kishor PB. 2018. The Emergence of Blockchain Technology and its Impact in Biotechnology, Pharmacy and Life Sciences. *Current Trends in Biotechnology and Pharmacy.*, 12(3):304–10.

Courtney JF, Paradise DB, Brewer KL, Graham JC. 2010. *Database Systems for Management*. 3rd edition. The Global Text Project.

Dowell RD, Jokerst RM, Day A, Eddy SR, Stein L. 2001. The distributed annotation system. *BMC Bioinformatics.*, 2:7.

Edgar F. Codd https://en.wikipedia.org/wiki/Edgar_F._Codd

Fleurence RL, Curtis LH, Califf RM, Platt R, Selby JV, Brown JS. 2014. Launching PCORnet, a national patient-centered clinical research network. *J Am Med Inform Assoc JAMIA.*, 21(4):578–582.

Fortier PJ, Michel HE. 2003. *Computer Data Processing Hardware Architecture*. In: *Computer Systems Performance Evaluation and Prediction*. Elsevier, p. 39–106.

Hellerstein JM, Stonebraker M, Hamilton J. 2007. Architecture of a database system. *Found Tren Databases.*, 1(2):141–259.

Johnson A, Pollard T, Shen L et al. 2016. MIMIC-III, a freely accessible critical care database. *Sci Data* 3., 160035.

Karsch-Mizrachi I, Takagi T, Cochrane G. 2018. International Nucleotide Sequence Database, C The international nucleotide sequence database collaboration. *Nucleic Acids Res.*, 46:D48–51.

Kleinaki A-S, Mytis-Gkometh P, Drosatos G, Efraimidis PS, Kaldoudi E. 2018. A blockchain-based notarization service for biomedical knowledge retrieval. *Comput Struct Biotechnol J.*, 16:288–97.

Kozomara A, Griffiths-Jones S. 2014. MiRBase: annotating high confidence microRNAs using deep sequencing data, *Nucleic Acids Res.*, 42: D68-73.

Lapatas V, Stefanidakis M, Jimenez RC, Via A, Schneider MV. 2015. Data integration in biological research: an overview. *J Biol Res (Thessalon).*, 22(1):9.

Lastdrager E. 2011. *Securing Patient Information in Medical Databases* [Internet]. University of Twente;. Available from: https://essay.utwente.nl/61035/1/MSc_E_Lastdrager_DIES_CTIT.pdf

Marshall J, Chahin A, Rush B. 2016. Review of clinical databases. In: *Secondary Analysis of Electronic Health Records*. Cham: Springer International Publishing;, 9–16.

Nguyen KA. *Database System Concepts*. OpenStax CNX; 2009 [cited 2021 Jan 29]. Available from: <http://cnx.org/contents/b57b8760-6898-469d-a0f7-06e0537f6817@1>

Ogasawara O, Kodama Y, Mashima J, Kosuge T, Fujisawa T. 2020. DDBJ database updates and computational infrastructure enhancement. *Nucleic Acids Res.*, 48:D45–50.

Okuda S, Yamada T, Hamajima M, Itoh M, Katayama T, Bork P, et al. 2008. KEGG Atlas mapping for global analysis of metabolic pathways, *Nucleic Acids Res.*, 36: W423-426.



2019-1-BG01-KA203-062371

- Oliveira AL. 2019. Biotechnology, big data and artificial intelligence. *Biotechnol J.*, 14(8):e1800613.
- Pollard T, Dernoncourt F, Finlayson S, Velasquez A. 2016. Data Preparation. In: *Secondary Analysis of Electronic Health Records*. Cham: Springer International Publishing;, 101–14.
- Ponten F, Schwenk JM, Asplund A, Edqvist PH. 2011. The Human Protein Atlas as a proteomic resource for biomarker discovery, *J Intern Med.*, 270: 428-446.
- Quek XC, Thomson DW, Maag JL, Bartonicek N, Signal B, Clark MB, et al. 2015. IncRNAdb v2.0: expanding the reference database for functional long noncoding RNAs, *Nucleic Acids Res.*, 43, D168-173.
- Rose PW, Beran B, Bi C, Bluhm WF, Dimitropoulos D, Goodsell DS, et al. 2011. The RCSB Protein Data Bank: redesigned web site and web services, *Nucleic Acids Res.*, 39: D392-401.
- Sayers EW, Beck J, Bolton EE, Bourexis D, Brister JR, Canese K, et al. 2021. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, 49(D1):D10–7.
- Schuler G.D., Epstein J.A., Ohkawa H., Kans J.A. 1996. Entrez: molecular biology database and retrieval system. *Methods Enzymol.*, 266:141–162.
- The RNACentral Consortium, RNACentral: an international database of ncRNA sequences. 2015. *Nucleic Acids Res.*, 43: D123-129.
- Watt A, Eng N. Types of Data Models. In: Watt A, Eng N, editors. *Database Design - 2nd edition*. BCcampus; 2014 [cited 2021 Jan 29]. Available from: <https://opentextbc.ca/dbdesign01>
- Watt A. Characteristics and Benefits of a Database. In: Watt A, Eng N, editors. *Database Design - 2nd edition*. BCcampus; 2014 [cited 2021 Jan 29]. Available from: <https://opentextbc.ca/dbdesign01/>
- Watt A. Data Modelling. In: Watt A, Eng N, editors. *Database Design - 2nd edition*. BCcampus; 2014 [cited 2021 Jan 29]. Available from: <https://opentextbc.ca/dbdesign01>
- Watt A. The Entity Relationship Data Model. In: Watt A, Eng N, editors. *Database Design - 2nd edition*. BCcampus; 2014 [cited 2021 Jan 29]. Available from: <https://opentextbc.ca/dbdesign01>
- Watt A. The Relational Data Model. In: Watt A, Nelson E, editors. *Database Design - 2nd edition*. BCcampus; 2014 [cited 2021 Jan 29]. Available from: <https://opentextbc.ca/dbdesign01>
- Zou D, Ma L, Yu J, Zhang Z. 2015. Biological databases for human research. *Genomics Proteomics Bioinformatics.*, 13(1):55–63.
- Zuniga PCC, Zuniga RAC, Mendoza MJ-A, Cariaga AA, Sarmiento RF, Marcelo AB. 2020. Workshop on Blockchain Use Cases in Digital Health. In: *Leveraging Data Science for Global Health*. Cham: Springer International Publishing;, 99–107.



Project website: www.digit-biotech.eu

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.