

DIGIT~Bio~TECH



ОРБ ОБЩОДОСТЪПНИ НАУЧНИ РЕСУРСИ: ДИГИТАЛНИ БАЗИ ДАННИ

Ниво за напреднали

АВТОРИ:

ФАБИАНО ШАЛХУБ И ЗОИ ГЕОРГИУ



Съдържание

Общодостъпни научни ресурси	4
Усъвършенствана структура на база данни	4
Системи за управление на база данни	4
Езици за модели на база данни	4
Характеристики на база данни	5
Типове модели на база данни	5
Фази на проектиране на база данни	6
Степени на абстракция	7
Схеми на база данни	7
Външно ниво	7
Концептуално ниво	7
Вътрешно ниво	8
Независимост на данните	8
Релационен модел на данни	8
Основни елементи на релационния модел на данни	9
Характеристики на таблицата	9
Диференциращи особености на релационния модел за база данни	10
Модел „обект-връзка“	10
Примери за обектни типове и връзки в биологичните бази данни	11
Модификационни аномалии	12
Ключови дефиниции	12
Език за структурирани заявки (SQL)	13
SQL команди	13
Платени и безплатни бази данни използвани в реалния живот	18
Платени бази данни	18
SAP HANA	18
IBM Db2 база данни	19
Oracle база данни	20
Безплатни бази данни	21



2019-1-BG01-KA203-062371

MySQL.....	21
PostgreSQL	22
Microsoft SQL	22
MariaDB.....	23
Oracle	23
Firebirdsql	24
Базите данни в модела на модерната наука	24
Преглед на базите данни в научната област	24
ДНК бази данни.....	26
РНК бази данни	26
Белтъчни бази данни.....	27
Бази данни за заболявания	27
Бази данни за експресия	27
Бази данни за пътища	27
Бази данни с медицинска информация за интензивно лечение (Medical Information Mart for Intensive Care, MIMIC)	30
PCORnet	31
Open NHS	31
Де-идентификация на базата данни.....	31
Приложение на блокчейн технологията в дигиталното здравеопазване	32
Литература	34



Общодостъпни научни ресурси

УСЪВЪРШЕНСТВАНА СТРУКТУРА НА БАЗА ДАННИ

Тази част разглежда усъвършенствания дизайн на базата данни. Обяснява структурата на базата данни и как да бъдат създадени връзки между таблиците в базата данни. Този материал представя и специфичния език, използван за отправяне на заявки (SQL) при извличането на данни от базата данни.

СИСТЕМИ ЗА УПРАВЛЕНИЕ НА БАЗА ДАННИ

Съвременната база данни може да бъде определена като структурирана колекция от информация (данни), която е представителна за реалния свят. Системите за управление на базата данни (СУБД, DBMS) се използват за създаване, управление и отправяне на заявки към базата данни. Понастоящем, реляционните системи за управление на бази данни (РСУБД, RDBMS) са най-развитите и широко използвани системи за бази данни в практиката. Почти всички онлайн транзакции и повечето онлайн системи за управление на съдържанието (напр. блогове и социални мрежи) разчитат на тези типове системи от бази данни, които са основни за световната инфраструктура за приложения. Фокус на СУБД е предоставянето на услуги, осигуряващи постоянство на данните в базата данни и функционалност, гарантираща че информацията е коректна и съгласувана, и че транзакциите следват свойствата на АСИТ (ACID) модела. АСИТ се отнася до четирите основни свойства на транзакцията:

- Атомност
- Съгласуваност
- Изолация
- Трайност

ЕЗИЦИ ЗА МОДЕЛИ НА БАЗА ДАННИ

Всички бази данни имат език за специфициране на структурата и съдържанието на базата данни. Специфицирането е вид проектна схема и представлява логически изглед на информацията, която се управлява от определена СУБД. Езикът за спецификация на базата данни трябва да бъде гъвкав, което да позволи да бъде използван дълго. Най-значимият елемент от базата данни, които се установява от специалистите по бази данни и разработчиците на приложения, е езика за манипулиране на данни. Той може да бъде под различна форма, като най-често срещаната е с интерфейс подобен на езиците за програмиране. Днес текстовите и



2019-1-BG01-KA203-062371

процедурни езици, такива като езика за структурирани заявки (Structured Query Language, SQL) и езика за заявки на обект (Object Query Language, OQL), остават едни от най-широко използваните форми за манипулиране на данни.

ХАРАКТЕРИСТИКИ НА БАЗА ДАННИ

Базата данни може да бъде определена като съгласувана, логична и вътрешно съгласувана. Освен това може да бъде определена и като самоописваща се, тъй като включва и метаданни, които от своя страна определят и описват данните и взаимовръзките между таблиците в базата данни. Те са проектирани да съдържат данни с определена цел. Всеки елемент от данни се съхранява в поле; съвкупността от полета образуват таблица. В една база данни може да има множество таблици.

За разлика от файл-базираната система, структурата на данните в системата на базата данни, се съхранява в системен каталог, а не в програмите на приложенията. Разграничаването между програми и данни се нарича „независимост на програмата от данните“.

Архитектурата на системата на базата данни се състои от набор от услуги, които по своята същност представляват основни услуги на операционната система, услуги за съхранение на системни файлове и услуги за управление на първичния буфер на паметта. Този набор от услуги се състои от: управление на каталога, управление на интегритета, управлението на транзакциите, контрол на конкурентността, управление на „заклучването“ (lock), управление на „мъртвата хватка“ (deadlock), управление на възстановяването, управление на сигурността, обработка на заявки, управление на комуникациите и управление на дневника (лога, log).

ТИПОВЕ МОДЕЛИ НА БАЗА ДАННИ

Моделите на данни могат да бъдат разделени на два типа:

- Високо ниво концептуални модели на данни
- Логически модели на данни базирани на записи

Високото ниво концептуални модели на данни предлагат възможност за представяне на данните по начин, подобен на начина, по който хората възприемат данните. Пример за такъв модел на данни е модела „същност-връзка“ (entity-relationship, ER), който се основава на понятия като същност (обект), атрибути и връзки (релации). Същността съответства на обект от реалния свят, атрибутите представляват свойствата на същността, а връзката показва взаимоотношенията между отделните същности.



2019-1-BG01-KA203-062371

Логическите модели на данни базирани на записи предлагат концепции, които потребителите могат да разберат и се основават на начина, по който данните се съхраняват в компютъра. Релационните модели на данни, мрежовия модел на данни и йерархичния модел на данни са трите най-разпространени логически модели на данни базирани на записи.

- При релационния модел, данните са представени под формата на релации, или таблици.
- При мрежовия модел, данните са представени като типове записи. Този модел представя също тип множество, дефинирано като ограничен тип от вида едно-към-много.
- При йерархичния модел, данните са представени като дървовидна йерархична структура, всеки клон на която представлява редица свързани записи.

ФАЗИ НА ПРОЕКТИРАНЕ НА БАЗА ДАННИ

Моделирането на данните представлява първата стъпка при проектирането на базата данни. Понякога обаче тази стъпка е фаза на високо ниво на абстрактност, известна още като концептуален модел. Тази фаза има за цел да опише:

- Данните представени в базата данни
- Връзката между елементите от данни
- Ограниченията на данните

На този първоначален етап от проектирането на базата данни, от съществено значение е да бъде направен анализ на изискванията към информацията. Това е най-важната фаза, тъй като цялостната ефективност на системата зависи от първоначално посочените изисквания към информацията и представата на потребителите за нея. Спецификата на изискванията към информацията направени на този етап определя крайната форма и съдържание на системата от база данни.

След като спецификациите са определени и разработени, те трябва да бъдат структурирани в интегрирана, свързана система, процедура означавана като логически дизайн. Логическият дизайн включва следните стъпки:

- i) разработване на модел на данни за представата на всеки потребител
- ii) интегриране на обектите, атрибутите и релациите в сложна логическа схема, която описва базата данни за този модел с термини, които не са свързани с използвания софтуерен пакет
- iii) трансформиране на логическата схема в софтуерна схема изразена чрез езика на изборния пакет за управление на базата данни

Последната стъпка в проектирането на базата данни е физическия дизайн. При тази стъпка, софтуерната схема се промени във форма, която може да бъде съвместима със специфичния



2019-1-BG01-KA203-062371

хардуер, операционната система и системата за управление на базата данни на конкретната организация. Във физическия дизайн се включват и прилагат изискванията за интегритет и сигурност и дизайна на навигационните пътища.

СТЕПЕНИ НА АБСТРАКЦИЯ

Абстракция на данните представлява скриване на определени подробности относно начина, по които данните се съхраняват и поддържат. По отношение степента на абстракция, моделите бази данни могат да бъдат разделени на три нива, а именно:

- Външно, или потребителско ниво, което е най-високото ниво на абстракция и представлява само част от цялата база данни
- Логическо ниво, което описва какви данни се съхраняват в цялата база данни
- Физическо ниво, което е най-ниското ниво на абстракция и описва как данните се съхраняват в базата данни

СХЕМИ НА БАЗА ДАННИ

Схема на база данни може да бъде дефинирана като описание на база данни на ранен етап, което не се очаква да бъде често променяно. В системата на базата данни съществуват множество схеми. Архитектурата на базата данни съдържа три нива на схеми.

Външно ниво

Това е най-висшето ниво схеми. Видът на външното ниво на данни се определя от специфичните приложения за обработка на данни или потребителския изглед. Външното ниво съдържа няколко изгледа и представя фрагменти от действителната база данни. Всеки изглед се предлага на потребител или група потребители, като по този начин се улеснява взаимодействието между потребителя и системата.

Концептуално ниво

Това ниво описва логическата структура на цялата база данни, която от своя страна се описва от прости логически концепции, включващи обекти, техните свойства или връзки. Следователно, сложността на детайлите за изпълнение на данните няма да бъдат видими за потребителите. В базата данни се поддържа само един концептуален изглед. За да могат обекти или атрибути да бъдат посочени в системата на базата данни, те първо трябва да бъдат дефинирани в изгледа на концептуалното ниво, описано още като логическа схема. Този изглед



2019-1-BG01-KA203-062371

на нивото трябва да бъде изключително стабилен, тъй като се приема за основа за разработването на външните и вътрешните нива на изгледа.

Вътрешно ниво

The way the data are stored and the way to access the data are described in this schema. The internal level represents the internal or physical state of the database. Its objective is to increase the efficiency of the database system, while fulfilling the required needs.

НЕЗАВИСИМОСТ НА ДАННИТЕ

Независимост на данните се отнася до способността на потребителските приложения да остават незасегнати от промените направени в дефинициите и организацията на данните. Съществуват два типа независимост на данните: логическа и физическа.

Логическа независимост на данните е способност да се променя логическата (концептуална) схема без това да засяга външната схема, или потребителския изглед. Корекциите в логическата схема, такива като промени в структурата на базата данни чрез добавяне на таблици, не би трябвало да променят функциите на приложенията (външни изгледи).

Физическа независимост на данните представлява способност на концептуалната схема да остане непроменена от промените направени във вътрешната схема. Промените във файловата организация или структурите за съхранение, устройствата за съхранение или стратегията за индексирание не водят до промени на концептуално ниво.

РЕЛАЦИОНЕН МОДЕЛ НА ДАННИ

Релационния модел на данни е разработен от д-р Едгар Ф. Код през 1970 год. Той представя данните в таблична форма, тъй като този начин за представяне е познат за много хора. В този модел се основава от логическата простота на плоските файлови структури. Релационният модел се основава на теорията на множествата, която дава възможност да бъдат извършвани няколко операции въз основа на релации. Този модел осигурява най-гъвкав достъп до данните, което го прави подходящ за използване в динамична среда при вземане на решения.

SQL е език за релационна трансформация; той предлага начини за формиране на релации и манипулиране на данни. Резултат от трансформацията винаги е друга релация, която може да съдържа само един ред и колона.



2019-1-BG01-KA203-062371

Основни елементи на релационния модел на данни

Таблица 1. Основни елементи на релационния модел на данни

Компонент на база данни	Описание
Таблица	включва колони и редове; подмножество на Декартово произведение на множество домейни, характеризиращо се с име
Колони	основни единици за съхранение; съдържат основни елементи на данните, на които съдържанието може да бъде разделено
Редове	съдържат свързани колони; заедно с колоните представляват основата на всяка база данни
Домейн	набор от приемливи стойности, които могат да бъдат включени в колона
Степен	брой колони в таблицата

Релацията, наричана още таблица или файл, може да бъде определена като двуизмерна таблица, съдържаща данни за класа на обекта или взаимовръзката между класовете обекти. Във всеки ред от таблицата се включват данни, отнасящи се до конкретен обект, а във всяка колона се включва конкретен атрибут. Редовете, или записите, на релацията се наричат още кортежи. Записът в таблицата представлява отделен обект. Броят редове в една релация е показател за нейната кардиналност. Броят на колоните, известни като полета или атрибути, в една релация определят на степента на тази релация. Основните елементи на релационния модел на данни са представени в таблица 1. Унарната релация съдържа само един атрибут; бинарната релация съдържа само два атрибута; тройната релация съдържа само три атрибута.

ХАРАКТЕРИСТИКИ НА ТАБЛИЦАТА

- Всяка таблица в базата данни има уникално име
- Няма дублирани редове; всеки ред е различен
- Всеки ред има различно име
- Последователността на редовете и колоните не е важна
- Записите в колоните произлизат от един и същ домейн в съответствие с типа на техните данните, включително: дата, логичност (истина/лъжа), символ (низ) и число (числово, цяло, с плаваща запетая ...)



2019-1-BG01-KA203-062371

ДИФЕРЕНЦИРАЩИ ОСОБЕНОСТИ НА РЕЛАЦИОННИЯ МОДЕЛ ЗА БАЗА ДАННИ

Същност: Структура от данни се счита за значима, ако нейното отстраняване води до загуба на информация в базата данни.

Правила за интегритет: Те гарантират, че съдържанието на базата данни остава точна и съгласувана. Има два типа интегритет:

1. Интегритет на обекта: Позволява уникална идентификация на всеки обект в реляционната база данни. Това позволява достъп до всички данни. Необходимо е първичния ключ да не е с нулеви стойности.
2. Референциален интегритет: Позволява преpraщане на кортежи чрез прилагане на външни ключове. Необходимо е стойностите приемани от външния ключ да съвпадат с тези на първичния ключ на базата данни или да имат нулеви стойности.

Манипулиране на данни: Метод за манипулиране на данни; основен подход за създаване на информация при вземане на решения.

МОДЕЛ „ОБЕКТ-ВРЪЗКА“

Моделът на данни „обект-връзка“ (entity-relationship, ER) е разработен преди повече от 35 години. Той е сравнително по-слабо абстрактен и лесен за обяснение. ER моделите лесно могат да бъдат преобразувани в отношения и да бъдат представени като диаграми. Моделът се основава на обекти и взаимовръзките между тях. „Обект“ в модела може да бъде физически съществуващ обект или такъв, който е само концептуален (идеен). Ако таблиците на обекта са зависими от неговото реалното съществуване, то този обект се означава като слаб. И обратното, ако обектът може да съществува независимо от всички свързани с него други обекти, тогава този обект е силен.

Съществуват различни видове обекти:

- Независими обекти или ядра: „Строителните блокчета“ на базата данни. Това са силни обекти. Първичният ключ не е външен ключ и може да бъде прост или съставен. Различните видове ключове са описани в таблица 2.
- Зависими или производни обекти: Те са зависими от съществуването на две или повече таблици. Те се използват за обединяване на две ядра и могат да включват други атрибути. Всяка свързана таблица се идентифицира с външен ключ. Възможни са три опции за първичния ключ: i) ако е уникален, се използва комбинация външни ключове на свързани таблици, ii) използване на външни ключове и квалифицираща колона, или iii) създаване на нов прост първичен ключ.
- Характеризиращи обекти: Тези обекти предлагат допълнителна информация за друга таблица. Те описват други обекти и са представителни за многозначни обекти. Външният ключ



2019-1-BG01-KA203-062371

се използва за последваща идентификация на характеризиранията таблица. Възможни са две опции за първичния ключ: i) използване на комбинация от външни ключове и квалифицираща колона, или ii) създаване на нов прост първичен ключ.

Таблица 2. Видове ключове

Видове ключове	Описание
Кандидат ключ	прост или съставен ключ, който е уникален, тъй като няма два реда в една таблица, които да имат една и съща стойност по една и също време, и минимален, тъй като всяка колона е необходима за постигане на уникалност
Съставен ключ	трябва да бъде минимален; съставен от два или повече атрибута
Първичен ключ	кандидат ключ избран от дизайнера на базата данни за използване като механизъм при идентифициране на целия набор от обекти; трябва уникално да идентифицира кортежи в таблицата и не може да бъде нула; в ER модела се идентифицират чрез подчертаване на атрибута
Вторичен ключ	атрибут използван само за извличане; може да бъде съставен
Алтернативен ключ	всички кандидат ключове, които не са избрани за първичен ключ
Външен ключ	атрибут в таблица, който препраща към първичен ключ в друга таблица ИЛИ може да бъде нулев

Нулева стойност: Различна от нула или празна стойност; не зависи от типа данни. Нулева стойност означава, че или действителната стойност не се знае, или че атрибута не е приложим.

ПРИМЕРИ ЗА ОБЕКТНИ ТИПОВЕ И ВРЪЗКИ В БИОЛОГИЧНИТЕ БАЗИ ДАННИ

Обектният тип описва характеристики, които се споделят от колекция обекти в един домейн. Например, „протеин“ може да се приеме като обектен тип с атрибути като секвенция, име, молекулна маса, видове и номер на присъединяване. Един обектен тип може да бъде представен по няколко начина като всеки от тях показва стойности на атрибута, които съответстват на конкретния обектен тип. Например, имената на два примера за обектния тип „протеин“ са човешки α -хемоглобин и китов миоглобин. Стойностите на техния атрибут „вид“ са съответно „човешки“ и „китов“.



2019-1-BG01-KA203-062371

Връзките показват, че два или повече обектни типове са свързани. Например, „протеин“ може да взаимодейства с много други „протеин“ обектни типове, или може да бъде член на семейство. Различните категории връзки могат да опишат природата на връзката. Например, един обектен тип може да бъде представен като част от друг (напр. „бета верига“ е част от „лист“ във вторичната структура на протеина) или като разновидност на друг (напр. „ензим“ е разновидност на „протеин“).

МОДИФИКАЦИОННИ АНОМАЛИИ

По време на процесите на вмъкване, изтриване или модифициране на данните в базата данни могат да възникнат неволни грешки. Ако грешката е резултат от дизайна на базата данни, тогава тази грешка се означава като модификационна аномалия.

Съществуват три типа модификационни аномалии:

1. Аномалии при изтриване: премахването на един логически обект води до загубата на информация за друг несвързан с него логически обект.
2. Аномалии при вмъкване: вмъкването на данни за един логически обект налага вмъкването на данни за друг несвързан логически обект.
3. Аномалии при актуализиране: промяната на информацията за един логически обект налага повече от една промяна в релацията.

КЛЮЧОВИ ДЕФИНИЦИИ

Централизирана база данни: данните в тази система се съхраняват на един сайт.

Разпределена база данни: базата данни и софтуера на СУБД са разпределени на различни сайтове, които са свързани чрез компютърна мрежа.

База данни: споделена колекция от свързани данни използвана за подпомагане работата на организациите.

Език за дефиниране на данни (Data Definition Language, DDL): използва се за дефиниране на концептуални и вътрешни схеми.

Система за управление на база данни (СУБД, Database Management System, DBMS): компютърни програми използвани за създаване, управление и отправяне на заявки към базата данни.

Модел на данни: колекция от концепции използвани за описание на структурата на базата данни.



2019-1-BG01-KA203-062371

Излишък на данни: съхранение на едни и същи данни на две или повече места в базата данни.

Нормализация: метод за структуриране на данни с цел намаляване или избягване на проблеми.

Възстановяване: процедура за използване на логове или резервни копия с цел пресъздаване на повредена база данни.

ЕЗИК ЗА СТРУКТУРИРАНИ ЗАЯВКИ (SQL)

SQL означава Език за структурирани заявки (Structured Query Language, SQL) и представлява компютърен език за съхранение, манипулиране и извличане на данни съхранени в реляционна база данни. Това е най-широко използвания език за бази данни. Той предлага начини за създаване на релации и манипулиране на данни. SQL е стандартен език за реляционни бази данни. Всички реляционни системи за управление на бази данни (PCYБД, RDMS), като MySQL, MS Access, Oracle, Sybase, Informix, Postgres и SQL Server, използват SQL като стандартен език за база данни, въпреки че използват и различни „диалекти“:

- MS SQL Server използва T-SQL
- Oracle използва PL/SQL
- MS Access използва версия на SQL наречена JET SQL (естествен формат) и др.

SQL команди

SQL командите покриват всички необходими действия извършани в базата данни. Въпреки това, както бе споменато по-горе, могат да съществуват някои различия между различните типове бази данни, включително и използването на тези „диалекти“. Всяка SQL команда притежава синтаксис и описание.

Командите в SQL се наричат „заявки“ и биват два типа:

1. Заявка за дефиниране на данни: дефинира структурата на базата данни, създава таблици, специфицира техните ключове, индекси и т.н.
2. Заявки за манипулиране на данни: това са заявки, които могат да бъдат редактирани.



2019-1-BG01-KA203-062371

SQL команди ¹:

Команда	Синтаксис	Описание
ALTER table	ALTER TABLE table_name ADD column_name datatype;	Използва се за добавяне на колона към таблица в база данни
AND	SELECT column_name(s)FROM table_nameWHERE column_1 = value_1 AND column_2 = value_2;	Оператор използван за комбиниране на две условия
AS	SELECT column_name AS 'Alias'FROM table_name;	Ключова дума в SQL използвана за преименуване на колона или таблица чрез прилагане на „алтернативно“ име
AVG	SELECT AVG(column_name)FROM table_name;	Използва се за агрегиране на числова колона и връщане на средната ѝ стойност
BETWEEN	SELECT column_name(s)FROM table_nameWHERE column_name BETWEEN value_1 AND value_2;	Оператор използван за филтриране на резултатите в определен диапазон
CASE	SELECT column_name,CASEWHEN condition THEN 'Result_1'WHEN condition THEN 'Result_2'ELSE 'Result_3'ENDFROM table_name;	Изявление използвано за получаване на различни резултати в изявлението SELECT
COUNT	SELECT COUNT(column_name)FROM table_name;	Функция, приемаща името на колоната като аргумент и отчита броя на редовете, когато колоната не е NULL
Create TABLE	CREATE TABLE table_name (column_1 datatype, column_2 datatype, column_3 datatype);	Използва се за създаване на нова таблица в базата данни и специфицира името на таблицата и колоните в нея
DELETE	DELETE FROM table_nameWHERE some_column = some_value;	Използва се за премахване на редове от таблица

¹ Източник <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;	Клауза в SQL използвана за агрегатни функции заедно с изявлението SELECT
HAVING	SELECT column_name, COUNT(*)FROM table_nameGROUP BY column_nameHAVING COUNT(*) > value;	Използва се в SQL тъй като командата WHERE не може да бъде приложена в агрегатните функции
INNER JOIN	SELECT column_name(s)FROM table_1JOIN table_2 ON table_1.column_name = table_2.column_name;	Използва се за комбиниране на редове от различни таблици ако условието JOIN е TRUE
INSERT	INSERT INTO table_name (column_1, column_2, column_3) VALUES (value_1, 'value_2', value_3);	Използва се за добавяне на нови редове в таблица
IS NULL/ IS NOT NULL	SELECT column_name(s)FROM table_nameWHERE column_name IS NULL;	Оператор използван с клаузата WHERE за проверка на празните стойности
LIKE	SELECT column_name(s)FROM table_nameWHERE column_name LIKE pattern;	Специфичен оператор използван с клаузата WHERE за търсене на специфичен модел в колона
LIMIT	SELECT column_name(s)FROM table_nameLIMIT number;	Клауза, определяща максималния брой редове, който трябва да съдържа резултата
MAX	SELECT MAX(column_name)FROM table_name;	Функция, която приема броя на колоните като аргумент и връща най-голямата стойност сред тях
MIN	SELECT MIN(column_name)FROM table_name;	Функция, която приема броя на колоните като аргумент и връща най-малката стойност сред тях
OR	SELECT column_nameFROM table_nameWHERE column_name = value_1 OR column_name = value_2;	Оператор използван за филтриране на резултатите така, че те да съдържат само редове, в които всяко условие е TRUE



2019-1-BG01-KA203-062371

ORDER BY	SELECT column_nameFROM table_nameORDER BY column_name ASC DESC;	Клауза използвана за подреждане на резултатите от определена колона по числен или азбучен ред
OUTER JOIN	SELECT column_name(s)FROM table_1LEFT JOIN table_2 ON table_1.column_name = table_2.column_name;	Използва се за комбиниране на редове от различни таблици дори ако условието е NOT TRUE
ROUND	SELECT ROUND(column_name, integer)FROM table_name;	Функция, която приема името на колоната и цялото число като аргумент и закръгля стойността на колоната до броя на десетичните знаци посочени в цялото число
SELECT	SELECT column_name FROM table_name;	Изявление използвано за извличане на данни от база данни
SELECT DISTINCT	SELECT DISTINCT column_nameFROM table_name;	Използва се, за да се посочи, че изявлението е заявка, която връща уникална стойност в определена колона
SUM	SELECT SUM(column_name)FROM table_name;	Функция използвана за връщане на сума от стойности от определена колона
UPDATE	UPDATE table_nameSET some_column = some_valueWHERE some_column = some_value;	Използва се за редактиране на редове в таблица
WHERE	SELECT column_name(s)FROM table_nameWHERE column_name operator value;	Клауза, използвана за филтриране на резултати, така че да съдържат само редове, в които условието WHERE е TRUE
WITH	WITH temporary_name AS (SELECT *FROM table_name)SELECT *FROM temporary_nameWHERE column_name operator value;	Използва се за съхранение на резултати от определена заявка във временни таблици чрез прилагането на алтернативни имена



2019-1-BG01-KA203-062371

Команди и синтаксис за заявки за данни от единична таблица или множество таблици ²:

Single Table	Multiple Table
SELECT c1 FROM t Избор на данни от колона c1 от таблица t	SELECT c1, c2 FROM t1 INNER JOIN t2 on condition Избор на колони c1 и c2 от таблица t1 и извършване на вътрешно съединение между t1 и t2
SELECT * FROM t Избор на всички редове и колони от таблица t	SELECT c1, c2 FROM t1 LEFT JOIN t2 on condition Избор на колони c1 и c2 от таблица t1 и извършване на ляво съединение между t1 и t2
SELECT c1 FROM t WHERE c1 = 'test' Избор на данни в колона c1 от таблица t, където c1=test	SELECT c1, c2 FROM t1 RIGHT JOIN t2 on condition Избор на колони c1 и c2 от таблица t1 и извършване на дясно съединение между t1 и t2
SELECT c1 FROM t ORDER BY c1 ASC (DESC) Избор на данни в колона c1 от таблица t във възходящ или низходящ ред	SELECT c1, c2 FROM t1 FULL OUTER JOIN t2 on condition Избор на колони c1 и c2 от таблица t1 и извършване на пълно външно съединение между t1 и t2
SELECT c1 FROM t ORDER BY c1LIMIT n OFFSET offset Пропускане изместването на редовете и връщане на следващите n реда	SELECT c1, c2 FROM t1 CROSS JOIN t2 Избор на колони c1 и c2 от таблица t1 и получаване на Декартово произведение от редове в таблица
SELECT c1, aggregate(c2) FROM t GROUP BY c1 Групиране на редове чрез агрегатна функция	SELECT c1, c2 FROM t1, t2 Избор на колони c1 и c2 от таблица t1 и получаване на Декартово произведение от редове в таблица
SELECT c1, aggregate(c2) FROM t GROUP BY c1HAVING condition Групиране на редове чрез агрегатна функция и филтриране на тези групи чрез клаузата 'HAVING'	SELECT c1, c2 FROM t1 A INNER JOIN t2 B on condition Избор на колони c1 и c2 от таблица t1 и съединението им чрез прилагане на клауза INNER JOIN

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



2019-1-BG01-KA203-062371

Платени и безплатни бази данни използвани в реалния живот

• 4th Dimension	• Google Fusion Tables	• MonetDB	• SAND CDBMS
• 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
• Alibase	• 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 Lotus Approach	• OpenLink Virtuoso (Open Source Edition)	• SQLBase
• Clarion	• IBM DB2 Express-C	• OpenLink Virtuoso Universal Server	• SQLite
• ClickHouse	• Infobright	• OpenOffice.org Base	• Scream DB
• Clustrix	• Informix	• Oracle	• SAP Advantage Database Server (formerly known as Sybase Advantage Database Server)
• CockroachDB	• Ingres	• Oracle Rdb for OpenVMS	• Teradata
• CSQL	• InterBase	• Panorama	• Tiberio
• CUBRID	• InterSystems Caché	• Paradox	• TIDB
• DataEase	• InterSystems IRIS Data Platform	• Percona Server for MySQL	• TimesTen
• DataFlex	• LibreOffice Base	• Percona XtraDB Cluster	• Trafodion
• Database Management Library	• Linter	• Pervasive PSQL	• Unisys RDMMS 2200
• Dataphor	• MariaDB	• Polyhedra	• UniData
• dBase	• MaxDB	• PostgreSQL	• UniVerse
• Derby (aka Java DB)	• Microsoft Access	• Postgres Plus Advanced Server	• Vectorwise
• Empress Embedded Database	• Microsoft Jet Database Engine (part of Microsoft Access)	• Progress Software	• Vertica
• Exasol	• Microsoft SQL Server	• Raima Database Manager (RDM)	• VoltDB
• EnterpriseDB	• Microsoft SQL Server Express	• R-Base	• YugabyteDB
• eXtremeDB	• SQL Azure (Cloud SQL Server)	• RethinkDB	
• FileMaker Pro	• Microsoft Visual FoxPro		
• Firebird	• Mimer SQL		
• FrontBase			

Фигура 1: Непълен списък на съществуващите бази данни

Тази част разглежда базите данни на пазара, независимо дали са безплатни или частни. Съществуват обаче толкова много бази данни (фигура 1), че ние не можем да споменем всички. Избрали сме и сме представили по-долу само „най-популярните“ и „най-често използваните“.

ПЛАТЕНИ БАЗИ ДАННИ

От огромния брой бази данни на пазара, ние избрахме да представим три от най-често използваните от големите компании и организации платени бази данни.

SAP HANA

Тази база данни е проектирана от европейската компания SAP SE, основана в Германия. SAP HANA е база данни, която е колоно-ориентирана и може да обработва SAP и не-SAP данни. Тя е проектирана да пази и извлича данни от приложения и други източници от множество нива на съхранение. SAP HANA може да бъде внедрена локално или в облак от редица доставчици на облачни услуги. Тази база





2019-1-BG01-KA203-062371

данни се използва от организации, които извличат данни от приложения и не са ограничени от бюджет.

Нейни основни характеристики са:

- Поддържа SQL, OLTP и OLAP.
- Намалява изискванията за ресурси чрез компресия.
- Данните се съхраняват в памет, което в някои случаи значително намалява времето за достъп.
- Възможни са отчети и управление на инвентара в реално време.
- Може да взаимодейства с множество други приложения.

От януари 2021, поддържаните хардуерни платформи ³ за SAP HANA са:

- Intel-базирани хардуерни платформи
- IBM Power Systems

От януари 2021, поддържаните операционни системи ⁴ за SAP HANA са:

- Linux SUSE
- Linux Red Hat

IBM Db2 база данни

IBM Db2 базата данни е основана в началото на 70-те години на миналия век, когато Едгар Ф. Код, изследовател, работещ за компанията, описва теорията за релационните бази данни и през юни 1970 год. публикува модел за манипулиране на данни. Днес, това е база данни, която има възможност да използва NoSQL, а може да чете и JSON ⁵ и XML файлове ⁶.



Текущата версия на DB2 е LUW 11.1, която предоставя множество подобрения. Едно от тях, по-специално, е усъвършенстването на BLU Acceleration (BLink Ultra или Big Data, Lightning fast и Ultra-easy), което има за цел да ускори работата на базата данни чрез използване на технология за пропускане на данни. Тази технология е разработена с цел увеличаване на скоростта на системи с повече данни, отколкото могат да бъдат побрани в тяхната памет.

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

⁴ За повече информация относно поддържаните операционни системи за SAP HANA, виж SAP Note 2235581 - SAP HANA: <https://service.sap.com/sap/support/notes/2235581>

⁵ JavaScript Object Notation е отворен стандартен файлов формат като XML и се счита за неструктурирани данни.

⁶ XML е отворен стандартен файлов формат като и се счита за неструктурирани данни.



2019-1-BG01-KA203-062371

Последната версия на Db2 също така осигурява подобрени функции за възстановяване при произшествия, съвместимост и анализи.

Нейни основни характеристики са:

- BLU Acceleration може да използва максимално наличните за огромни бази данни ресурси.
- Може да е в облак, физически сървър или и в двете по едно и също време.
- Чрез използването на Task Scheduler, могат да се изпълняват едновременно множество задачи.
- Кодовете за грешка и кодовете за изход, чрез Task Scheduler, могат да определят кои задачи да бъдат изпълнявани.

Поддържаните хардуерни платформи ⁷ за IBM Db2 към януари 2021 г. са:

- IBM z/Architecture mainframe
- Intel-базирани хардуерни платформи

Поддържаните операционни системи за IBM Db2 към януари 2021 г. са:

- z/OS
- Unix
- Linux
- Windows

Oracle база данни

Oracle база данни обикновено се използва за извършване на онлайн обработка на транзакции (online transaction processing, OLTP) и съхранение на данни (data warehousing, DW). Тя може също да комбинира операции по OLTP и DW. Oracle базата данни може да бъде разположена локално, в облак или да бъде представена като хибридна облачна инсталация. Тя може да функционира на сървъри на трети страни, както и на локален хардуер на Oracle Exadata, на Oracle Cloud или на частен облак на клиента.

Първата версия е пусната през 1979 г. като нейното развитие е повлияно от изследванията на Едгар Ф. Код върху дизайна на релационните бази данни.

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



2019-1-BG01-KA203-062371

Нейни основни характеристики са:

- Това е междуплатформена база данни. Тя може да работи на различен хардуер в операционните системи, включително Windows Server, Unix и различни GNU/Linux.
- Има свой мрежови стек, позволяващ на приложения от различни платформи да комуникират безпроблемно с Oracle базата данни, напр. приложения работещи под Windows могат да се свързват с Oracle база данни работещи под Unix.
- Това е база данни съвместима с АСИТ, която поддържа интегритета и надеждността на данните.

Понастоящем поддържаните хардуерни платформи са:

- Proprietary Oracle Database Appliance
- Sparc
- IBM Power Systems
- X64- базирани хардуерни платформи

Понастоящем поддържаните операционни системи ⁸ са:

- Unix
- Linux
- Windows

БЕЗПЛАТНИ БАЗИ ДАННИ ⁹

Ако базата данни е безплатна, това не означава непременно, че не се начисляват такси на потребителите. Това е вярно за повечето бази данни, но някои разработчици избират да ограничат определени функции и налагат такса за използването на тези функции (вижте първа част на Основно ниво).

MySQL

MySQL е общодостъпна реляционна база данни, която работи на различни операционни платформи, включително Windows, Linux, macOS, и др. Облачната версия на MySQL може да бъде използвана за пакетирани софтуер, важни за бизнеса системи и големи по обем уебсайтове.



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

⁹ Източник <https://www.guru99.com/free-database-software.html> актуализиран през 2021



2019-1-BG01-KA203-062371

Нейни основни характеристики са:

- Осигурява мащабиране и гъвкавост
- Трайност на съхранението на данни в мрежата и хранилището
- Висока производителност
- Има стабилна поддръжка на транзакциите

PostgreSQL

PostgreSQL е корпоративен клас общодостъпна система за управление на база данни. Тя функционира както с SQL за релационни, така и с JSON за не-релационни заявки. Системата се поддържа от опитни разработчици, които имат огромен принос тя да бъде високонадежден софтуер за управление на данни. Работи на три различни платформи, а именно Windows, Linux и macOS. Не е налична нейна облачна версия. Чрез PostgreSQL е възможно създаването на персонализирани типове данни и различни методи за заявки. Съхранената процедура може да функционира под различни програмни езици.



Нейни основни характеристики са:

- Съвместима с различни платформи, използващи всички основни езици и мидълуер.
- Сървър в режим на готовност и лесна достъпност
- Притежава функционалност за програмиране от страна на сървъра
- Лог-базиран и тригер-базиран SSL за репликация
- Предлага най-сложния заключващ механизъм
- Поддръжка на контрол на конкурентността в множество версии
- Поддръжка на клиент-сървър мрежова архитектура
- Обект-ориентирана и ANSI-SQL2008 съвместимост
- PostgreSQL позволява връзка с други бази данни като NoSQL, който функционира като обединен център за многоезични бази данни.

Microsoft SQL

SQL Server е PCУБД разработена от Microsoft. Тя поддържа ANSI SQL, който е стандартен SQL език. SQL Server обаче притежава своя собствена версия на SQL езика, T-SQL (Transact-SQL). Той функционира на Docker Engine, Ubuntu, SUSE Linux Enterprise Server и Red Hat Enterprise Linux. Налична е и облачна версия.





2019-1-BG01-KA203-062371

Нейни основни характеристики са:

- Осигурява интеграция на структурирани и неструктурирани данни чрез SQL Server и Spark.
- Предлага мащабиране, изпълнение и наличност за критично важни интелигентни приложения, хранилища на данни и езера с данни.
- Предлага разширени възможности за защита на данните.
- Достъп до обширни, интерактивни Power BI отчети при вземането на по-бързи и по-правилни решения.

MariaDB

MariaDB е клон на MySQL системата за управление на база данни. Тя е създадена от оригиналните разработчици на MySQL. Тази СУБД дава възможност за обработка както на малки, така и на големи, корпоративни задачи. Тя функционира на три платформи, а именно Windows, Linux и macOS. Налична е и нейна облачна версия. MariaDB е алтернативен на MySQL софтуер. Тя предлага високо степен на мащабиране посредством лесно интегриране.



Нейни основни характеристики са:

- Функционира под GPL, BSD или LGPL лицензи.
- Разполага с много начини за съхранение, включително такива които са високо производителни и могат да бъдат интегрирани с други РСУБД.
- Предлага Galera клъстерна технология.
- MariaDB работи на различни операционни системи и поддържа множество програмни езици.

Oracle

Oracle е самоподдържаща се, самозащитаваща се и самоуправляваща се база данни, проектирана да елиминира ръчното управление на данните. Тя е интелигентна, сигурна и лесно достъпна база данни в облак, която подпомага развитието на бизнеса. Работи на две платформи, Windows и Linux. Налична е и облачна версия.



Нейни основни характеристики са:

- Oracle Cloud е оптимизирана за високопроизводителна работа с бази данни, стрийминг и работа с хипер големи данни.
- Можете лесно да мигрирате в облака.
- Предлага услуги в зависимост от вашите предпочитания за работа, като може да стартира Oracle Cloud във вашия център с данни.



2019-1-BG01-KA203-062371

Firebirdsql

Firebird е общодостъпна SQL РСУБД, която работи с Microsoft Windows, macOS, Linux и някои Unix платформи, включително HP-UX, Solaris и AIX. Налична е като облачна версия. Firebird има лесна за развитие езикова поддръжка, съхранени процедури и тригери.



Нейни основни характеристики са:

- Firebird позволява да създадете своя персонализирана версия.
- Безплатна за изтегляне, регистрация и използване.
- Притежава подобрена мултиплатформена РСУБД.
- Предлага редица възможности за финансиране, от членство към Ffirebird до спонсорство.

Базите данни в модела на модерната наука

Този раздел е посветен на общодостъпните бази данни използвани в науката, как да ги използваме и да се възползваме от наличните знания.

ПРЕГЛЕД НА БАЗИТЕ ДАННИ В НАУЧНАТА ОБЛАСТ

Съществуващи научни бази данни и как да ги използваме.

Както бе споменато по-рано, споделянето, интегрирането и аотирането на данни е важна част от биологичните изследвания, тъй като позволява на изследователите да възпроизведат експериментите и да интерпретират получените научни резултати. Макар да се счита че биоинформатиците и компютърните специалисти са отговорни за съхранението на данните, биолозите имат еднаква с тях роля за насърчаване интеграцията на данни, тъй като те са тези, които генерират този тип данни и обикновено са крайни потребители.

Интегрирането на данни представлява процес на комбиниране на данни от различни източници с цел предлагане на потребителя на единна представа за тези данни. В компютърната наука, теоретичните рамки за интегриране на данни са категоризирани въз основа на метода приложен за тяхното интегриране и биват „нетърпеливи“ и „мързеливи“. Според нетърпеливия метод, познат още като складиране, данните се копират в глобална схема и се съхраняват в централно хранилище на данни. Терминът “схема” означава организиран и „работещ със заявки“ подход за съхранение на данни. При мързеливия метод, данните се съхраняват на различни места и се интегрират при поискване в съответствие с глобалната схема за разпределение на данните



2019-1-BG01-KA203-062371

между тези места. Обемът на данните, техният собственик и съществуващата структура са основните фактори, определящи кой от двата метода ще бъде използван при интегрирането на данните. Освен това, в биологичните науки, тези методи могат да бъдат прилагани по различен начин и да бъдат използвани на различни нива. В резултат на това са били формулирани шест различни, широко използвани схеми за интегриране на данни:

- Централизиране на данни: данните се намират в централизирани ресурси. UniProt и GenBank са два примера за бази данни, използващи този метод.
- Съхранение на данни: данни от различни източници се намират в едно централно хранилище. Pathway commons е база данни, използваща този подход за интегриране на данни.
- Интегриране на набор от данни: вътрешни работни заявки достигат до разпръснати бази данни и изтеглят данни към локалното хранилище.
- Хипервръзки: този подход позволява на потребителите да имат достъп до бази данни и инструменти от различни области на природните науки, като по този начин се насърчава оперативната съвместимост между различните научни области. ExPASy е типичен пример за портал, базиран на този тип методология за интегриране на данни.
- Обединени бази данни: тъй като произхождат от хетерогенни източници, при интегрирането на този тип данни е необходимо те да бъдат предварително „преведени“. Това означава, че данните се трансформират в общоприет формат, така че да могат да бъдат интерпретирани по същия начин при тяхното последващо интегриране. Типичен пример за този тип е Distributed Annotation System (DAS), която представлява клиент-сървърна система.
- Свързани данни: Мрежа от взаимосвързани онлайн достъпни данни. Огромната система на Linked Data е изградена от множество графични потребителски интерфейси (graphical user interfaces, GUI), състоящи се от хипервръзки и свързващи асоциирани данни от множество доставчици на данни. BIO2RDF е типичен пример за база данни, която използва този подход като основа при интегрирането на данни.

Според използвания подход за интегриране на данни, централизирането на данни, съхранението на данни и интегрирането на набор от данни се основават на „нетърпеливата“ теоретична рамка, докато хипервръзките, обединените бази данни и свързаните данни се основават на „мързеливата“ теоретична рамка.

Форматът на данни представлява организиран начин за представяне на данни и метаданни под формата на файл. Учените започнаха да съхраняват биологичните данни във форматирани файлове, тъй като непрекъснатото нарастване на количеството данни породи необходимостта те да бъдат анализирани посредством компютърни системи и бази данни. Проблемът, който възниква във връзка с форматирани данни, е свързан с появата на множество формати дори при представянето на един и същ тип данни. В някои случаи е установено, че за представянето на данни и метаданни в един файл могат да бъдат използвани повече от един клас формати. Най-широко използваните класове формати са: i) таблици, ii) FASTA-подобни, iii) tag-



2019-1-BG01-KA203-062371

структурирани, и iv) GenBank-подобни. Най-доброто решение на този проблем би било учените да се съгласят да използват ограничен брой специфични формати, което би довело до опростяване на процеса на интегриране на данни. Разработването на конвертори, които да трансформират различните класове формати също би било едно подходящо решение.

Според непълния списък на списание Nucleic Acids Research, понастоящем в употреба са повече от 1700 биологични бази данни. За да бъдат използвани за специфична научна цел, данните съхранявани в базите данни трябва да бъдат интегрирани и структурирани. Съществуващите биологични бази данни се състоят от информация за широк спектър от изследвания, такива като геномика на безгръбначни, белтъчни секвенции, човешки гени и заболявания, ДНК нуклеотидни секвенции, клетъчна биология, имунология, метаболитни и сигнални пътища, протеомика и др.

Както бе споменато в Основно ниво, класифицирането на биологичните бази данни зависи от няколко фактора, включително обхвата на данните и нивото на биокурация. Независимо от това, най-простия и всеобхватен начин за класифициране на базите данни е въз основа на типа данни. Следователно, в следващия раздел базите данни ще бъдат представени като ДНК, РНК, белтъчни, такива за заболявания, за експресия и за пътища.

ДНК бази данни

ДНК базите данни са фокусирани върху обработката на ДНК данни от конкретен или различни видове. Основна цел на човешката ДНК база данни е да определи референтен геном, да направи профилиране на генетичните вариации при човека, да свърже генотипа с фенотипа и да идентифицира метагеномите в човешките микробиоми. Пример за ДНК база данни е GenBank, общодостъпна колекция от всички изследвани ДНК секвенции. Към февруари 2021 година, в GenBank са налични повече от 776 милиарда нуклеотидни двойки (<http://www.ncbi.nlm.nih.gov/genbank/statistics>).

РНК бази данни

Тези бази данни включва информация за не-кодиращи РНКи (ncRNAs), такива като микроРНКи и дълги не-кодиращи РНКи (lncRNAs), които не кодират белтъци. Целта на РНК база данните е да декодира не-кодиращи РНКи, от които дългите не-кодиращи РНКи са най-често изучаваните, и да опише техните функции и взаимодействия. Пример за такава РНК база данни е RNACentral, която съдържа унифицирани данни за не-кодиращи РНКи секвенции получени от различни бази данни, сред които Rfam, miRBase и lncRNAdb.



2019-1-BG01-KA203-062371

Белтъчни бази данни

Белтъчните бази данни са разработени с цел създаване на обширна колекция от данни за белтъци, идентифициране на белтъчни семейства и домейни, реконструкция на филогенетични дървета и профилиране на белтъчни структури. Типичен пример за този тип бази данни е PDB базата данни, която съдържа структурата на хиляди белтъчни макромолекули.

Бази данни за заболявания

Базите данни за заболявания включват информация за различни типове заболявания, но най-вече са фокусирани върху представянето на данни относно различните видове рак. Един от най-важните проекти, посветени на рака, е разработен от The Cancer Genome Atlas (TCGA), чиято цел е да събере широк набор от омикс данни, информация за иРНК, еднонуклеотидния полиморфизъм (SNP) и метилирането за над двадесет различни вида рак при човека.

Бази данни за експресия

Базите данни за експресия могат да бъдат използвани за различни цели, такива като изучаване на тъканно-специфичната гена експресия и регулация, съхранение на експресионни данни, установяване на диференцираща и базова експресия, и изследване и преглед на информацията за експресия, получена от различни РНК и белтъчни бази данни. Като база данни за експресия, Human Protein Atlas включва експресионните профили на значителна част от човешките протеин-кодиращи гени, получени от РНК и белтъчните бази данни.

Бази данни за пътища

Базите данни за пътища включват данни за биологичните пътища, които могат да бъдат използвани от изследователите за анализ на метаболитни, регулаторни и сигнални взаимодействия. Типичен пример за база данни за пътища е KEGG PATHWAY, която съдържа информация за молекулни взаимодействия и реакционни мрежи.

Националният център за биотехнологична информация (National Center for Biotechnology Information, NCBI), част от Националната медицинска библиотека на САЩ към Националния институт за здравеопазване, е разработил интегрирана система за извличане на данни, която предлага достъп до 34 различни бази данни, общо 3.0 милиарда записа, наречени Entrez. Глобалната страница за търсене на Entrez (<https://www.ncbi.nlm.nih.gov/search/>) има връзки към всяка една от 34 бази данни. Системата Entrez е лесна за използване тъй като позволява на потребителите да изтеглят данни в различни формати и да търсят текст като използват прости булеви заявки. Записите в различните бази данни са свързани помежду си и могат да бъдат представени в различни формати. Освен това, потребителите на Entrez имат възможност да изтеглят единични записи или група от записи. Някои от 34 бази данни, които са част от Entrez, са: PubMed (<https://pubmed.ncbi.nlm.nih.gov/>), която съдържа научни и медицински



2019-1-BG01-KA203-062371

абстракти/цитати; BioSample (<https://www.ncbi.nlm.nih.gov/biosample>), съдържаща информация за източници на биологични материали; GEO Profiles (<https://www.ncbi.nlm.nih.gov/geoprofiles>), включваща профили за гена експресия и молекулното многообразие и dbVar (<https://www.ncbi.nlm.nih.gov/dbvar>), съдържаща данни от изследвания върху структурните вариации на генома.

Данните, съхранявани в NCBI произхождат от три източника: i) директно от изследователи, ii) национални и международни партньорства или споразумения с доставчици на данни или научни консорциуми, и iii) вътрешно куриране. Трябва да се отбележи, че NCBI отговаря за управлението на GenBank базата данни и участва в Международната колаборация за база данни на нуклеотидни секвенции (International Nucleotide Sequence Database Collaboration, INSDC) в сътрудничество с Европейския архив за нуклеотиди EMBL-EBI (European Nucleotide Archive, ENA) и Японската ДНК базата данни (DNA Data Bank of Japan, DDBJ).

Тъй като базите данни са се доказали като полезен инструмент в много научни области, тяхното използване в областта на здравеопазването непрекъснато набира популярност. Днес, технологичният напредък в областта за работа с данни позволява на здравните специалисти да събират, обработват и анализират данни за здравето, което води не само до подобряване качеството на здравните услуги, но и до повишаване сигурността на пациентите и потребителите. За да бъдат направени тези подобрения, съответните данни трябва да се събират, съхраняват и анализират по подходящ и сигурен начин и да се обменят между отделните нива на обслужване в здравната система. Това доведе до разработване на Електронни здравни досиета (ЕЗД; Electronic Health Records, EHRs), бази данни, които съхраняват данни за пациентите и могат да бъдат достъпни и използвани от здравните специалисти.

Електронните здравни досиета могат да бъдат определени като медицински бази данни, които предлагат достъп на потребителите, в случая здравни специалисти и административен персонал, до здравните досиета на пациентите. Най-разпространените ЕЗД са електронното медицинско досие (ЕМД; Electronic Medical Record, EMR) и лично здравно досие (ЛЗД; Personal Health Record, PHR). Електронните медицински досиета съдържат информация предоставена от отделно болнично отделение, цяла болница или част от болница. Те могат да съдържат и информация от много болници. Информация от този тип обикновено се добавя в ЕЗД само от медицински персонал. От друга страна, личните здравни досиета се попълват от пациентите. ЛЗД представлява електронно приложение, което осигурява на пациентите сигурна платформа, на която те да могат да контролират и споделят данни за своето здраве. Основната разлика между двата типа ЕЗД е тази, че ЛЗД трябва да бъде представено по начин разбираем от пациента, докато ЕМД прилича на своя хартиен вариант, тъй като те са достъпни само за здравни специалисти.

Първата ЕЗД система става достъпна през 60-те години на миналия век, като нейното създаване бе провокирано от натрапването на неструктурирана и неизползвана информация за пациенти в продължение на няколко десетилетия. Големи организации започнаха да създават системи от бази данни с цел съхранение и структуриране на данни в централизирани хранилища. Тези бази данни позволяват организирането и колекционирането на данни от много различни



2019-1-BG01-KA203-062371

източници, включително аптеки, лаборатории, клинични изследвания и звена за клинична грижа, и такива като записи за администриране на лекарства. Понастоящем, ЕЗД системата се прилага предимно в развитите страни. Например, Закона за здравни информационни технологии за икономическо и клинично здраве (Закон НІТЕСН от 2009 г.) доведе до дигитализиране на системата за предоставяне на здравни грижи в САЩ и последващо развитие на програмите Medicare и Medicaid за стимулиране използването на ЕМД.

Основна цел за създаването на ЕЗД бе необходимостта от структуриране и архивиране досиетата на пациентите. По-късно те бяха развити и като средство за заплащане на здравни услуги и подобряване на качеството на обслужването. През годините, с напредването на технологиите, ЕЗД стават все по-широко обхватни, динамични и взаимосвързани. Независимо от това, за разлика от другите индустрии, големите бази данни не се използват така широко в медицинската сфера. Това се дължи на ниското качество на събраните данни и тяхната лоша структурираност. Преди разработването на ЕЗД, медицинските изследвания се основаваха на регистрите на заболяванията или на системите за управление на хроничните заболявания (chronic disease management systems, CDMS). Тези хранилища имат значителни ограничения, тъй като се състоят от колекции от данни, които често се отнасят само към едно специфично заболяване. Освен това те не могат да трансформират данните или да правят връзки към други заболявания, тъй като съдържат информация само от група пациенти от определена географска област. От друга страна, ЕЗД данните са разнообразни, което улеснява анализирането на комплексни клинични взаимовръзки и вземането на решения.

Компонентите на ЕЗД могат да бъдат от различен тип медицински данни и да варират от здравни досиета до необработени сензорни данни. Медицинските данни се разделят на прецизни и непрецизни данни. Прецизните данни включват информация за пациента или такава, която може да бъде свързана с конкретен пациент. Непрецизните данни включват сензорни данни, наричани още данни от измервания, тъй като включват само данни от сензори, такива като данни от ЕЕГ. Данните, съхранявани в медицинските бази данни се означават като метаданни. Най-често използваният тип бази данни за съхранение на медицинска информация са релационната база данни, която представя данните под формата на таблици, състояща се от редове и определен брой колони. Някои бази данни могат да включват информация за пациента, такава като медицинска история на пациента, или анонимни данни, които могат да бъдат използвани при следващи проучвания.

Медицинските данни могат да бъдат разделени на няколко категории, описани по-долу:

- Медицински и лабораторни данни: Здравните специалисти могат да правят поръчки за медикаменти или лабораторни изследвания в система за въвеждане на лекарски заявки, които в последствие се изпълняват от лаборанти или медицински сестри. Примери за тази категория данни са рецепти за лекарства и микробиологични резултати.
- Данни за плащания: Тази категория медицински данни се състои от кодове използвани от болниците за подаване на искове към техните застрахователни



2019-1-BG01-KA203-062371

компания. Международната класификация на болестите, създадена от СЗО, и Настоящата терминология на процедурите, поддържана от Американската медицинска асоциация, са най-популярните системи за кодиране.

- Изображения: Това могат да бъдат рентгенографски изображения, получени от изследвания с рентген, ехокардиограми и компютърна томография (КТ).
- Бележки и отчети: Те могат да бъдат свързани с процеса на лечение на пациентите. Документите при хоспитализация също принадлежат към тази категория. Резултатите от образните изследвания обикновено се описват в оперативните доклади. Бележките трябва да бъдат частично структурирани чрез използването на система от шаблони.
- Физиологични данни: Тази категория медицински данни съдържа жизненоважни показатели, като сърдечна честота и кръвно налягане, както и ЕКГ и ЕЕГ диаграми.

Релационните бази данни са най-често използвания тип бази данни за управление и съхранение на медицински данни. Те могат да бъдат определени като колекция от таблици, които са свързани чрез споделени ключове. Схемата на базата данни определя начина, по който таблицата ще бъде структурирана и техните взаимовръзки. Една проста медицинска база данни може да съдържа четири таблици:

- Таблица 1: списък на пациентите
- Таблица 2: дневник за прием в болницата
- Таблица 3: списък с измервания на жизненоважни показатели
- Таблица 4: списък с кодове на жизненоважни показатели и свързани таблици

Първични и външни ключове могат да бъдат използвани за свързване на тези четири таблици.

Освен ползи от използването на базите данни в здравеопазването, те имат и някои ограничения, свързани най-вече с осигуряване на поверителност на данните и плановете за получаване на финансови приходи от данните. Независимо от това, за публично ползване са достъпни различни общодостъпни медицински бази данни, някои от които са описани по-долу.

Бази данни с медицинска информация за интензивно лечение (Medical Information Mart for Intensive Care, MIMIC)

Базата данни с медицинска информация за интензивно лечение, MIMIC, (<http://mimic.physionet.org>) е създадена през 2003 г. в резултат на сътрудничество между MIT, Philips Medical Systems и Beth Israel Deaconess Medical Center (BIDMC). Данните, въведени в тази база данни, са от пациенти приети в отделенията на BIDMC, на които са направени медицински и хирургични интервенции. Състои се от информация за над четири хиляди пациенти, детайлни физиологични и клинични данни, и е „деидентифицирана“ и свободно достъпна за



2019-1-BG01-KA203-062371

изследователите. В тази база данни присъстват два типа данни: клинични данни от ЕЗД, които се съхраняват в релационна база данни, съставена от 50 таблици, и резултати от легловите монитори, съхранявани в плоски бинарни файлове. Целта на тази колаборация е да създаде и оцени подобрени системи за мониторинг и лечение на пациенти, нуждаещи се от интензивни грижи, които ще направят процеса на вземане на решения в критични ситуации по-ефективен, по-бърз и по-точен.

PCORnet

Националната пациент-ориентирана мрежа за клинични изследвания (National Patient-Centered Clinical Research Network, PCORnet), е инициатива, която стартира през 2013 г., и чиято цел е да интегрира данни от няколко Мрежи за изследвания на клинични данни и Пациент-ориентирани изследователски мрежи. Тя съдържа 29 мрежи, които ще улеснят достъпа до огромно количество изследвания. Тя събира данни от рутинни посещения на пациенти при техните лекари, и данни, които се споделят от отделни пациенти чрез ЛЗД или от мрежи на пациентски общности.

Open NHS

Националните здравни служби (National Health Services, NHS) на Англия поддържат едно от най-големите в света хранилища, съдържаща данни относно здравето на хората. Open NHS¹⁰ е общодостъпна база данни, даваща достъп до информация, предоставена от правителството или други обществени организации. Този проект е създаден с цел повишаване на прозрачността и мониторинг на ефективността на Английските здравни служби. Дава се възможност на пациенти, здравни служители и проверители да сравняват качеството на здравните услуги и лечение в различни области на страната чрез използване на специално създадена за това база данни.

Де-идентификация на базата данни

Една от първите стъпки при създаването на база данни от ЕЗД е нейната де-идентификация. Преди базата данни да стане достъпна за ползване от изследователи и приложения, е необходимо да се гарантира спазването на политиките и разпоредбите за поверителност на данните. За структурирани данни, като колони в таблици, де-идентификацията е базирана на категоризирането на данните и последващо изтриване или криптографиране на тези, обозначени като поверителни. За неструктурирани данни, такива като документи при хоспитализация, се използват различни техники на изразяване, от такива като прости регулярни изрази до сложни невронни мрежи, които намират поверителната информация в свободния текст и я изтриват или криптографират.

¹⁰ Общодостъпните данни за NHS са от адрес: <http://www.england.nhs.uk/ourwork/tsd/data-info/open-data/>



2019-1-BG01-KA203-062371

ПРИЛОЖЕНИЕ НА БЛОКЧЕЙН ТЕХНОЛОГИЯТА В ДИГИТАЛНОТО ЗДРАВЕОПАЗВАНЕ

Блокчейн технологията се основава на наличието на децентрализирана система за съхранение на данни, където копие от регистъра на извършените транзакции се предоставя на всеки участник/разклонение. Това прави невъзможна промяната на данни от един участник без да бъдат уведомени останалите. Силно централизираните обекти биха имали значителна полза от използването на блокчейн технологията. Приложенията за дигитално здравеопазване до голяма степен зависят от централизираните системи. Следователно, блокчейн технологията притежава потенциал да трансформира дигиталното здравеопазване чрез промяна на начина на съхранение и защита на данните. Предложени са множество области за неговото използване, включително вериги за доставки, проверка на лекарства, възстановяване на искания, контрол на достъпа, и клинични изпитвания.

Установено е, че медицинските данни са най-високо ценени от хакерите, като според последни проучвания един медицински запис може да струва до 400 американски долара. Това означава, че защитата на данните в медицинските бази данни е от съществено значение. Блокчейн технологията предоставя решение на този проблем, като гарантира поверителност на данните чрез запазване на тяхната цялост, удостоверяване и упълномощаване при използването им. Данните в блокчейн са криптирани и ако някой иска да изтрие или повреди данните, това е възможно само чрез прилагането на механизъм за унищожаване на ключа, при който оригиналният ключ използван за първоначалното криптиране на данните се унищожават или стана безполезен. След това данните съхранявани в блокчейна няма да бъдат достъпни за четене.

Блокчейн технологията отговаря на две основни нужди по отношение на данните: запазване на интегритета и не-отхвърляне. Интегритет означава, че заявката и извлечените данни не могат да бъдат променяни, след като операцията по извличане е приключила. Не-отхвърлянето означава, че услугата по извличане на данни не може да отрече, че конкретните данни са били доставени от услугата в отговор на подадена заявка. Блокчейн технологията може да бъде определена като система за управление на разпределени транзакции, която не може да бъде нарушена. Тя може да се прилага за запазване на целостта на данните в системата на ЕЗД, споделянето и контрол на достъпа, съхранение и управление на данни.

Теоретичната блокчейн-базирана „нотаризирана“ заявка се състои от три изчислителни слоя:

- интерфейс на потребителя на данни
- интерфейс за комуникация с интерфейси на медицински бази данни, и
- механизъм на договора, който организира заявката и връща извлечените резултати към потребителя, извършва и подготвя транзакциите, и управлява договорите и метаданните



2019-1-BG01-KA203-062371

При прилагането на „нотаризирани“ заявки се използват две различни схеми: основна схема и версионизирана схема. Основната схема използва „заявка-отговор книга“, чрез която потребителят получава доказателство, че в определен момент конкретна заявка е достигнала до медицинската база данни, която от своя страна е изпратила исканите резултати. Тази схема може да бъде използвана, за да се гарантира интегритета и не-отхвърлянето на заявката, когато жизненоважна медицинска задача разчита на конкретна заявка. Версионизираната схема позволява нерегламентирано версионизиране (промяна) на данни, извлечени от динамично развиваща се медицинска база данни в различни моменти от време като използва една и съща заявка. Тази схема се използва за потвърждаване на различни варианти на променящи се медицински доказателства извлечени от често-актуализирани медицински бази данни.

Внедряването на блокчейн технологиите в областта на природните науки и фармацевтичните изследвания дава възможност за децентрализиране на интерфейса и споделянето на данни, което води до повишаване ефективността, бързината и неограничено мащабиране. Блокчейн осигурява непроменливост на данните, което е изключително важно при клиничните изпитвания, тъй като не позволява клиничните резултати да бъдат променени от изследователите в последствие. Тази технология може да бъде използвана и при идентификацията на лекарства, тяхното проследяване и верификация. Съществуват и определени рискове при внедряването на блокчейн технологията, които са свързани най-вече с осигуряване на поверителност, транзакции извън „веригата“ и съмнение относно възприемането на технологията от потребителите. Независимо от това, ползите от блокчейн технологията многократно надхвърлят нейните възможни недостатъци и играят съществена роля за ограничаване на незаконните действия с медицински данни.



2019-1-BG01-KA203-062371

Литература

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

Подкрепата на Европейската комисия за изготвянето на настоящата публикация не представлява одобрение на съдържанието, което отразява гледните точки само на авторите и не може да се търси отговорност от Комисията за всяка употреба, която може да бъде използвана за информацията, съдържаща се в нея.