

# Терминологичен англо - български речник по бази от данни

## А

**ACID** - основни свойства на транзакцията:

**Atomicity (атомарност, неделимост)** - транзакцията е неделима единица от работа. Тя или се изпълнява изцяло, или не се изпълнява въобще.

**Consistency (състоятелност, непротиворечивост)** - транзакцията преобразува базата данни от едно непротиворечиво състояние в друго.

**Independency (независимост)** - всяка транзакция се изпълнява независимо от другите. Резултатите от обновяване на базата остават невидими за останалите транзакции до приключване на изпълнението.

**Durability, persistence (устойчивост, необратимост)** - промените в състоянието на базата, в резултат от действието на транзакцията, са постоянни и необратими.

**Access path (път за достъп)** - структура, която позволява ефективно намиране на определени записи от базата данни.

**Aggregate functions (агрегатни функции)** - функции, които се прилагат върху всички стойности на някой атрибут от дадено отношение и връщат единствена стойност. В стандарт ISO са предвидени функциите: SUM, MAXIMUM, MINIMUM, AVERAGE и COUNT. Вж. също **Relation, Attribute**.

**Aggregation (агрегация)** - форма на абстракция, при която връзката между различни типове обекти се разглежда като групиран на по-високо ниво обект от нов тип. Моделират се връзки от вида "ЧАСТ-ОТ" (IS-A-PART-OF).

**Alias (синоним)** – алтернативно име на таблица. Използва се при заявки на SQL. Вж. също **SQL, Query**.

**Allocation (разположение)** - процес на избор на най-подходящ възел за разполагане на даден фрагмент от гледна точка на ефективно изпълнение на заявки и транзакции при разпределените бази от данни. Вж. също **Fragmentation**.

**Application Design (проектиране на приложението)** – етап от създаването на приложение от тип "база данни", който обхваща проектирането на потребителския интерфейс и на програмите за използване на базата.

**Application programmer (приложен програмист)** – лице, което реализира спецификациите, създадени от системния аналитик във вид на програми. Вж. също **System analyst**.

**Association (асоциация)** - форма на абстракция, при която връзката между подобни обекти се разглежда като обект-множество на по-високо ниво. Моделира се връзка от тип "ЧЛЕН-НА" (MEMBER-OF) между обекта-множество и неговите обекти-членове.

**Attribute (атрибут)** – в релационния модел това е именована колона на отношение; в E-R модела – свойство на обект или връзка.

**Authentication (удостоверяване)** – проверка правата за достъп на потребителя. Осъществява се чрез потребителско име и парола.

## В

**Big Data („големи данни“)** - информационни активи с голям обем, произлизащи от различни източници, които се създават, събират и обработват бързо. Данните са различни по тип: структурирани, полуструктурирани, неструктурирани. Изискват иновативни технологии за обработка и анализ. Целта е да се подпомага вземането на решения в дадена организация и да се осигури конкурентно предимство. Вж. също **NoSQL**.

**Binary Large Objects – BLOB (големи двоични обекти)** - съдържат двоични данни, представляващи изображение, дигитализирано видео и/или аудио, процедура или произволен неструктуриран обект. СУБД няма достъп до съдържанието на обекта или неговата вътрешна структура.

**Binary Relationship (двоична връзка)** – термин в E-R модела, чрез който се описва взаимодействие между два обекта, например СЛУЖИТЕЛ “РАБОТИ\_ЗА“ ОТДЕЛ.

**Bottom-up approach (подход “отдолу-нагоре”)** – подход за проектиране, при който схемата на конкретна релационна база от данни се разглежда като съвкупност от функционални и други типове зависимости, задавани върху атрибутите от проектанта на базата. След като всички зависимости се дефинират, с помощта на алгоритъм за нормализация се синтезират релационни схеми в трета нормална форма (3NF) чрез групиране на подходящи атрибути. Нарича се още релационен синтез. Вж. също **Functional dependency, Normalization, Third normal form**.

## С

**Candidate key (кандидат ключ)** – в релационния модел това е възможен ключ на дадена релационна схема. Вж. също **Relation schema**.

**Cardinality (мощност)** – брой на кортежите в отношението. Вж. също **Tuple, Relation**.

**Cardinality ratio (кардинално число)** – в E-R модела задава броя на състоянията на връзката, в които даден обект може да участва. Вж. също **Relationship occurrence**.

**Cartesian product (декартово произведение)** – множествена операция на релационната алгебра, която комбинира кортежи от две отношения.

**CASE tools (средства за автоматизиране на проектирането)** – средства, чрез които се подпомагат различните дейности, свързани със създаването на бази от данни.

**Casual user (инцидентен потребител)** – мениджъри на средно и високо ниво, които несистемно имат достъп до базата и се нуждаят всеки път от различна информация. Те използват сложни езици за заявки, за да оформят своите запитвания към базата.

**Centralized approach (централизиран подход)** – подход, при който се събират всички изисквания, преди да започне проектирането на концептуалната схема на базата.

**Class (клас)** - съвкупност от (тип на) обекти с еднаква структура и поведение.

**Classification (класификация)** - форма на абстракция, при която класът на обекта се дефинира чрез множество от обекти със сродни характеристики. Моделират се връзки от вида “Е-ПРЕДСТАВИТЕЛ-НА” (IS-AN-INSTANCE-OF) или “Е-ЧЛЕН-НА” (IS-A-MEMBER-OF). Вж. също **Relationship type**.

**Client (клиент)** – софтуерно приложение, което изисква определени услуги от един или няколко сървъра. От гледна точка на технологията “бази от данни” клиентът съдържа логиката на приложението и предоставя потребителски интерфейс. Вж. също **Server, Client-server**.

**Client-server (клиент-сървър)** – архитектура на многопотребителска СУБД, при която съществува процес, наречен “клиент”, който изисква някакъв ресурс и “сървър”, предоставящ този ресурс. Обикновено сървърът се разполага в един възел от локална мрежа, а клиентите – в други възли. Вж. също **Client, Server**.

**Clustering (кълъстериране)** – съвместно съхраняване на таблици с общи колони в една група (кълъстер). Общите колони образуват кълъстер-ключ, който се съхранява само на едно място и подобрява едновременно достъпа до групирани таблици.

**Cluster index (кълъстер индекс)** - главен индекс, ако полето, по което е подреден файла е с повтарящи се стойности. Вж. също **Index**.

**Column (колона)** – вж. също **Attribute**.

**Composite attribute (съставен атрибут)** - атрибут, изграден от други атрибути, всеки от които има независимо значение.

**Composite key (съставен ключ)** – кандидат ключ, изграден от няколко атрибута. Вж. също **Candidate key**.

**Conceptual modeling (концептуално моделиране)** - процесът на изграждане на база от данни, независимо от начина на нейното реализиране. Използват се обектно-базирани модели. Представява важен етап от проектирането на качествени бази от данни.

**Conceptual schema (концептуална схема)** - дефинира глобалната структура на базата от данни и описва цялостното информационно съдържание на базата. При нейното изграждане се използва модел на данните. Вж. също **Data model**.

**Concurrency control (контрол на паралелния достъп до данните)** – услуга на СУБД, която осигурява безконфликтната работа на много потребители с базата и запазва цялостността на данните. Вж. Също **Database Management System**.

**Consistency (състоятелност, непротиворечивост)** – вж. **Data integrity**.

**Constraint (ограничение)** – правило за непротиворечивост на базата, което не може да се нарушава.

Видове:

UNIQUE - полето в таблицата трябва да съдържа уникална стойност.

CHECK – полето може да съдържа само специфични типове от данни и позволени стойности.

DEFAULT – полето съдържа стойност по подразбиране.

PRIMARY KEY – главният ключ съдържа уникални стойности.

FOREIGN KEY – чуждият ключ трябва да съответства на съществуващ главен ключ в друга таблица.

## D

**Data (данни)** - отделните факти за различни обекти от реалния свят.

**Data abstraction (абстракция на данните)** - процес на идентифициране на основните аспекти на дадено понятие (обект) и игнориране на маловажните му свойства. Представява разширение на понятието "тип на данните". Позволява дефинирането на абстрактни операции върху абстрактни данни, без да се има предвид конкретно представяне чрез някакви структури. Вж. също **Information hiding, Encapsulation**.

**Data administration (администриране на данните)** - управлението на данните като организационен ресурс. Включва планиране на базата, разработване и поддържане на стандарти, политики и процедури, концептуално и логическо проектиране.

**Data administrator (администратор на данните)** - мениджър, отговарящ за управлението на данните като информационен ресурс, с оглед постигане целите на организацията.

**Data conversion and loading (конвертиране и зареждане на данни)** – фаза от жизнения цикъл на приложение от тип “база от данни, при която съществуващи данни се прехвърлят в новата база.

**Data Definition Language (DDL) (език за дефиниране на данните)** – език на СУБД, чрез който се задават структурата на базата и ограниченията за цялостност. Вж. също **Database Management System, Data integrity.**

**Data dictionary (речник на данните)** – вж. **System catalog.**

**Data independence (независимост на данните)** - разделяне описанието на данните от приложенията, които работят с тях. Това е възможността да се променя схемата на едно ниво на системата, без това да налага промяна в схемите на останалите, по-високи нива. Съществуват два вида независимост на данните: логическа и физическа.

**Data integrity (цялостност на данните)** – това е възможността да се работи с точни, актуални и непротиворечиви данни, независимо от разположението им във файловете във всеки един момент от време.

**Data management (управление на данните)** - процес на прилагане на информационни технологии, с цел да се преобразуват огромните масиви с данни на организацията в смислена информация, която да подпомага вземането на бизнес решения.

**Data Manipulation Language (DML) (език за обработване на данните)** – език на СУБД, чрез който се задават видовете обработки върху съхраняваната база. Вж. също **Database Management System.**

**Data Mining (извличане на данни)** - процес на автоматично извличане на знания от големи обеми данни във формата на шаблони, статистически правила или тенденции за подпомагане вземането на бизнес решения.

**Data model (модел на данните)** - съвкупност от абстрактни понятия, чрез които се описват свойствата на разработваното приложение (напр. информационна система).

**Data modelling (моделиране на данните)** - процес, при който на всяко приложение се съпоставя представяне на свойствата с отчитане на спецификата му и изискванията към обработките.

**Data redundancy (излишество при съхраняване на данните)** – дублиране на едни и същи данни в различни файлове.

**Data security (сигурност на данните)** – задават се права на потребителите и допустими операции. Това става чрез специален модул на СУБД за защита и контрол на достъпа до базата. Вж. също **Authentication.**

**Database (база от данни)** - множество от свързани данни, групирани за нуждите на даден потребител.

**Database administrator (DBA) (администратор на базата)** - специалист по информационни технологии, който отговаря за реализирането на дадена база, защитата на данните, ефективната работа на системата. Вж. също **Data administrator.**

**Database design (проектиране на базата)** - фаза от жизнения цикъл на приложение от тип “база от данни, при която се създава логически и физически проект на базата от данни като се отчитат характеристиките на избраната СУБД за реализация на приложението. Състои се от шест подфази.

**Database designer (проектант на базата)** - определя какви данни ще се съхраняват в базата при разработване на ново приложение.

**Database integrity (цялостност на базата)** – вж. **Data integrity**.

**Database Management System (DBMS) (система за управление на бази от данни)** - съвкупност от програми, които позволяват на потребителя да създава и обработва бази от данни, свързани с различни приложения. Вж. също **Database**.

**Database server (сървър на базата)** – вж. **Server**.

**Data striping (образуване на групи)** – техника за разпределяне на данни, до които е необходим едновременен достъп, върху няколко диска. Целта е да се уравни натоварването и паралелно да се изпълняват входно-изходните операции. Постига се намаляване на времето за обработка на заявките и се увеличава производителността на системата.

**Database system (система “бази от данни”)** - СУБД и управляваните чрез нея бази от данни. Вж. също **Database Management System, Database**.

**Database utility (помощна програма на базата)** - средство, което улеснява работата на администратора на базата. Вж. също **Database administrator**.

**DBMS selection (избор на СУБД)** – фаза на процеса “проектиране на базата”, която обхваща дейности, свързани с избора на софтуер от тип СУБД за реализация на разработваното приложение.

**Data Warehouse (хранилище с данни)** - множество от интегрирани, тематично ориентирани, устойчиви множества от данни, проектирани за поддържането на процеса “вземане на решения”, където всяка единица от данни е смислена в определен момент от време.

**Data Warehousing (изграждане на хранилище с данни)** - процес на интегриране на големите по обем корпоративни данни в единствено хранилище, което съдържа конгломерат от разнообразни по произход и тип данни. Процесът е стратегически важен за обединението на различните информационни източници в организацията. Вж. също **Data Warehouse**.

**Degree of a relationship (степен на връзката)** – брой на участващите във връзката типове обекти. Вж. също **Relationship type**.

**Delegation (делегиране)** - алтернативен на наследяването механизъм за споделяне на методи и функционалност между отделни класове обекти. При него не е задължително класовете непременно да са свързани в йерархия.

**Delete (изтриване)** – операция, която служи за изтриване на кортежи от дадено отношение.

**Denormalization (денормализация)** – процес, при който се променя структурата на дадена таблица в базата, за да бъде тя в по-ниска нормална форма с оглед подобряване производителността на системата. Вж. също **Normalization, Normal forms**.

**Derived attribute (изчислим атрибут)** - атрибут, чиято стойност може да се пресметне от стойностите на други атрибути.

**Design methodology (методология на проектиране)** - систематичен процес, обхващащ всички дейности свързани с проектирането на определена база от данни без оглед на модела на СУБД – релационен, обектно-ориентиран, обектно- релационен или друг.

**Difference (разлика)** - множествена операция на релационната алгебра, приложима върху две съвместими по обединение отношения. Резултатът съдържа кортежите, които фигурират само в едното отношение. Аналогична е на операцията “разлика” на две множества. Вж.също **Union compatible**.

**Dimension (измерение)** – категория, по която се подреждат данните (фактите). Обикновено се използват измеренията продукти, местоположение, време. Измерението притежава собствени атрибути и може да е изградено от няколко нива, като на всяко ниво данните са обобщени с различна степен на детайлизация. Всеки факт има определена мярка. Вж.също **Multidimensional Database**.

**Dimension table (таблица на измеренията)** - състои се от редове, всеки от които съдържа стойности на атрибутите на дадено измерение. Броят на таблиците с измерения съвпада с броя на конкретните измерения. Вж.също **Multidimensional Database**.

**Distributed processing (разпределена обработка)** - традиционна архитектура за многопотребителски СУБД, която се състои от компютър и определен брой терминали.

**Distributed database (разпределена база от данни)** - логически интегрирано множество от споделени между отделни потребители данни (и тяхното описание), физически разпределени по възлите на компютърна мрежа.

**Distributed DBMS (разпределена СУБД)** - софтуер, позволяващ управлението на разпределена база от данни, при което разположението на данните по възлите остава скрито за нейните потребители.

**Domain (домейн, област)** – множество от допустими стойности на един или няколко атрибута. Вж. също **Attribute**.

**Domain constraints (ограничения върху домейните)** - налагат неделимост на стойностите, изграждащи домейните. Това е основна характеристика на релационния модел. Вж. също **Domain**.

**Durability (устойчивост)** – вж. **Persistent data**.

## Е

**Encapsulation (капсулиране)** - метод за структуриране на една система. Представява свързването в едно на данни и процедури за тяхната обработка.

**End users (крайни потребители)** - лицата, които имат достъп до базата чрез терминали или работни станции, за да извличат необходимата им информация и да обновяват данните.

**Enterprise (фирма, предприятие)** - източник на данните.

**Entity (обект)** – първично понятие, отделни единици от реалността, за които се събират данни. Съвпада с **Object**.

**Entity integrity (ограничение за цялостност на обекта)** – в отношението не се допускат нулеви стойности на главния ключ. Вж. също **Primary key, Relation**.

**Entity occurrence (представител, екземпляр на типа обект)** – уникално идентифицируем обект, принадлежащ на определен тип. Вж.също. **Entity type**.

**Entity type (тип обект)** - множество от обекти с еднакви атрибути. Вж. също **Attribute**.

**Entity-relationship diagram (диаграма “обект-връзка”)** – средство в E-R модела за представяне схемата на базата от данни в графичен вид. Вж. също **Entity-relationship model**.

**Entity-relationship model (модел „понятие-взаимодействие“)** – концептуален модел, чрез който се представят обектите, връзките между тях и основните им свойства съгласно определени потребителски изисквания. Вж. също **Object type, Relationship type, Attribute**.

**Equijoin (съединение по еквивалентност)** – операция на релационната алгебра, вариант на операцията „съединение”, където в условието за съединение, допустимото сравнение е равенство. Вж. също **Join**.

**Extract, Transform, Load – ETL (извлечи, трансформирай, зареди)** – процес, при който данните се извличат от източници, стандартизират се по формат и се зареждат в хранилището. Вж. също **Data Warehouse**.

## F

**Fact table (таблица с факти)** - съдържа конкретните данни за даден бизнес процес в зависимост от броя на измеренията и указател коя стойност към кое измерение се отнася. Вж. също **Dimension, Dimension table**.

**Field (поле)** - съвкупност от символи или пространството, заделено за един информационен елемент. Това е най-малката единица от информация, до която СУБД осигурява достъп. Вж. също **Database Management System**.

**File (файл)** – именувана съвкупност от данни, съхранена върху носител. Съвкупност от записи.

**File organization (файлова организация)** - физическата организация на данните от файла в записи, блокове и структури за достъп. Вж. също **File**.

**File –server (файлов сървър)** - архитектура на многопотребителска СУБД, при която обработката е разпределена по мрежата. На файловия сървър се съхраняват файловете и СУБД, т.е. той представлява споделено дисково устройство. Вж. също **Database Management System**.

**File-based system (файлово-базирана система)** - множество от приложни програми, които доставят определени услуги чрез достъп до множество от файлове. Вж. също **File**.

**First normal form (1NF) (първа нормална форма)** – при нея отношението съдържа само неделими стойности, т.е. клетката, образувана от пресичането на произволен ред с произволна колона винаги съдържа единствена стойност. Тя забранява множество от стойности, редица от стойности и влагането на релации. Вж. също **Relation, Normalization, Normal forms**.

**Foreign key (чужд, външен ключ)** – колона в дадена таблица, която съвпада с кандидат-ключ на друга таблица. Вж. също **Table, Candidate key**.

**Fourth Generation language (4GL) (език от четвърто поколение)** - непроцедурен език, чрез който потребителят указва какви функции трябва да извършва разработваното приложение, без да указва как да се осъществи това.

**Fragmentation (фрагментиране)** - процес на разбиване на базата на логически единици (фрагменти), които се съхраняват на различни възли.

**Full functional dependency (пълна функционална зависимост)** – функционална зависимост, при която всеки атрибут зависи функционално от съставния ключ, но не зависи от негово подмножество. Вж. също **Functional dependency**.

**Functional dependency (функционална зависимост)** – описва връзки между колоните в дадена таблица. Например, колоната  $X$  функционално определя колоната  $Y$  (записва се  $X \rightarrow Y$ ) тогава и само тогава когато, за всеки кортеж, стойностите на  $Y$  се определят еднозначно (в зависимост) от стойностите на  $X$ .

## G

**Granularity (грануларност)** - големина на полето с данни, избрано от протокола за управление на едновременния достъп като единица за защита.

**Generalization (обобщение)** - форма на абстракция, при която подобни обекти се свързват с пораздащ ги обект на по-високо ниво, като се пренебрегват различията помежду им и се отделят само общите им белези. Моделират се връзки от тип „ТОВА-Е” (IS-A).

**Grouping (групиране)** – обособяване на кортежите в дадено отношение в групи в зависимост от стойностите на някой от атрибутите и прилагането на агрегатна функция в рамките на всяка група. Вж. също **Relation**.

## H

**Hash file (директен файл)** - файлова организация, при която местоположението на записите се изчислява с помощта на хеш-функция, която трансформира стойността на определено поле във физически адрес на блок.

**Heap file (неструктуриран файл)** - файлова организация, при която записите се поместват така както са въведени физически и не е установена наредба върху тях. Вж. също **Sequential file**.

## I

**Information hiding (скриване на информацията)** - принцип на софтуерното проектиране. Представлява отделяне на външните за даден обект аспекти (интерфейс) от вътрешните му детайли (реализация), които остават скрити и недостъпни за потребителя.

**Inheritance (наследяване)** - механизъм за споделяне на поведение и атрибути между отделни класове, които се намират в йерархична връзка помежду си.

**In memory database (база в паметта)** - СУБД, при която цялата база се съхранява в паметта на компютъра, а не на външна памет. Осигурява по-бързо изпълнение на заявките, защото липсва време за търсене.

**Implementation (реализация)** – фаза на процеса “проектиране на базата”, при която се реализира проектираната база, като се следва създаденото описание на схемата на езика за дефиниране на данните на избраната СУБД.

**Index (индекс)** - допълнителна структура към основния файл с данни, чиято цел е да ускори извличането на търсената информация. Вж. също **File, Access path**.

**Information (информация)** - данни, обработвани за определена цел; набор от факти, подредени по някакъв признак.

**Information system (информационна система)** - средство, което обхваща всички форми на събиране, съхраняване, извличане, обработка и разпространение на



информация. Тя е множество от взаимосвързани компоненти, което доставя информационни услуги и подпомага процеса на вземане на решения, координацията и контрола в рамките на дадена организация.

**Insert (въвеждане, вмъкване)** – операция, която служи за въвеждането на нов кортеж в дадено отношение.

**Instance (състояние)** - данните в базата във всеки един момент от време.

**Integrated data (интегрирани данни)** – данните се разглеждат като съвкупност от различни файлове, където няма повтарящи се полета.

**Integrity constraints (ограничения за цялостност)** - правила, които се извеждат от семантиката на данните и са присъщи на конкретно приложение. Проверяват се автоматично от СУБД, за да не се наруши състоятелността на базата. Вж. **Data integrity**.

**Internal schema (вътрешна схема)** - задава физическата структура на базата от данни, т.е. как се съхраняват данните като използва физически модел на данните. В нея се описват наредбата на полетата и записите, размерите, пътищата за достъп (указатели и индекси).

**Intersection (сечение)** - множествена операция на релационната алгебра, приложима върху две съвместими по обединение отношения. Резултатът съдържа кортежите, които фигурират и в двете, участващи отношения. Еквивалентна е на операцията „сечение” на две множества. Вж. също **Union compatible**.

**Index Sequential Access Method – ISAM (индексно последователен файл)** – статична файлова организация, при която файлът с данни се сортира по стойностите на избрано поле за търсене и върху това поле се дефинира главен индекс. Записите на файла се обработват последователно или поотделно. Достъпът до определен запис става с помощта на индекса, дефиниран върху стойностите на избраното поле за търсене. Вж. също **Index**.

## J

**Java Database Connectivity (JDBC)** – стандарт за достъп до база от потребителска програма, написана на езика Java. Вж. също **Client, Server, Client-server**.

**Join (съединение)** - операция на релационната алгебра, която комбинира в един кортеж свързани помежду си кортежи от две отношения. Двете отношения трябва да притежават общ атрибут. Операцията е ключова за всяка релационна СУБД, защото позволява обработката на връзки между отделните отношения.

## K

**Key (ключ)** – в релационния модел това е минималният суперключ на дадена таблица; в E-R модела – атрибут, по чиито стойности отделните екземпляри на даден тип обект се идентифицират еднозначно.

## L

**Logical database design (проектиране на логическата база)** - фаза на процеса “проектиране на базата”, която обхваща създаването на концептуална и външни схеми в термините модела на избраната за реализация СУБД.

## M

**Materialized view (материализирана представа)** – нова таблица в базата, която съдържа резултат от изпълнението на заявка, при която се извличат данни от няколко таблици.

**Meta-data (мета-данни)** - данни за самите данни, описание на всяка информационна единица в базата.. Вж. също **System catalog**.

**Multidimensional Database –MDB (многомерна база)** – база от данни с основна структура куб и няколко размерности. Използва се за анализ на данните в хранилището с данни и средствата за аналитична интерактивна обработка. Вж. също **OLAP Cube**.

**Multi-valued attribute (многостойностен атрибут)** – атрибут, който може да приема множество от стойности.

## N

**Natural join (естествено съединение)** - операция на реляционната алгебра, вариант на операцията „съединение по еквивалентност”, където от резултата са отстранени повтарящите се колони. Вж. също **Join**.

**Network-attached storage – NAS (памет, включена към мрежата)** - специализирано устройство, което осигуряват памет за споделяне на файлове. Позволява добавянето на дискови ресурси без да се налага изключване или обновяване на сървъри.

**Non-key attribute/column (неключов атрибут/колона)** – атрибут/ колона, която не участва в ключа.

**Normal forms (нормални форми)** - условия, които дадена реляционна схема трябва да удовлетворява. Първите три нормални форми - 1NF, 2NF и 3NF - са създадени от автора на реляционния модел Код. Могат да се разглеждат като фази на процеса „нормализация”. Вж. също **Normalization**.

**Normalization (нормализация)** - процес, при който неудовлетворителните ("лошите") реляционни схеми се разлагат или декомпонират чрез разбиване на атрибутите и групирането им в по-малки реляционни схеми, които притежават желани свойства. Нормализацията е формален метод за анализ на отношенията. Тя включва множество от правила, чрез които проектираните реляционни схеми се подлагат на проверки, за да се установи в коя от нормалните форми се намират. Вж.също **Normal forms**.

**NoSQL (не само SQL)** – модел, позволяващ съхраняването на неструктурирани данни – видео, изображения, съобщения и др.

**Null (празно поле)** – специална системна константа, чрез която се отбелязва, че за даден кортеж стойностите на някой от атрибутите са неопределени или атрибутът е неприложим. Различна е както от числото нула, така и от символа “интервал”, представляващи стойности от определен тип.

## O

**Object (обект)** - елемент от обкръжаващата ни действителност, за която се събират данни в базата. Съвпада с **Entity**.

**Object data base (база от обекти)** - множество от устойчиви и споделени между много потребители обекти, чието поведение, състояние и връзки се описват чрез обектен модел на данните.

**Object Data Management System (обектна система за управление на данни)** - съвкупност от програми, които позволяват дефинирането и обработката на обектна база.

**Online Analytical Processing – OLAP (аналитична интерактивна обработка)** – технология, позволяваща динамичен анализ, синтезиране и обединяване на големи обеми от многомерни данни. Вж. също **Multidimensional Database**.

**OLAP Cube (куб за аналитична интерактивна обработка)** – метод за съхраняване на данни в структура от тип многомерна матрица (куб с данни), подредена в зависимост от определени измерения. Използва се за генерирането на отчети с различно ниво на детайлност. Вж. също **Dimension**.

**Online Transaction Processing (OLTP) system (система, обработваща интерактивни транзакции)** - приложение, при което много потребители работят едновременно с една и съща база като паралелно изпращат различни заявки към нея.

**Open Database Connectivity (ODBC)** – стандарти за достъп до база от потребителска програма на клиента. Вж. също **Client, Server, Client-server**.

**Ordered file (сортиран файл)** - файлова организация, при която записите се подреждат по стойностите на едно или няколко от полетата на записа (сортиращи полета).

**Outer join (външно съединение)** - операция на релационната алгебра, вариант на операцията „съединение“, където в резултата фигурират и кортежи, за които не е намерен съответстващ. Бива ляво и дясно. Вж. също **Join**.

## Р

**Parametric user (параметричен потребител)** – краен потребител, който изпълнява еднотипни и стандартни заявки към базата.

**Parallel DBMS (паралелна СУБД)** – СУБД, която работи върху многопроцесорни компютри и осигурява управление на общите ресурси.

**Participation constraint (ограничение за участие)** - определя дали съществуването на един тип обект зависи от това, че е свързан с друг тип обект чрез някакъв тип връзка. Бива два вида: общо и частично.

**Partitioning (разделяне)** - метод за намаляване на натоварването в произволен хардуерен компонент на системата. Използва се при проектирането на физическата база.

**Persistent data (устойчиви данни)** - относително постоянни във времето факти, съхранени на някакъв носител. Премахването им от базата е явно и става чрез изпълнението на определена команда. Запазват се при един сеанс на работа с базата до следващ такъв.

**Physical database design (проектиране на физическата база)** - фаза на процеса “проектиране на базата”, при която се избират специфични структури на съхранение и пътища за достъп за файловете, изграждащи базата, за да се получи ефективно изпълнение на свързаните с нея приложения.

**Polymorphism (полиморфизъм)** - възможност да се изпращат съобщения на различни обекти, които да ги интерпретират специфично в зависимост от вида си.

**Primary index (главен индекс)** - индекс, дефиниран върху поле с уникални стойности, по което физически е подреден даден файл. Вж. също **Index**.

**Primary key (главен ключ)** - в релационния модел това е избраният кандидат-ключ на дадена релационна схема; в E-R модела – ключът на типа обект. Вж. също **Key**.

**Privileges (привилегии)** – позволените на даден потребител операции върху определена таблица от базата.

**Project (проекция)** - операция на релационната алгебра, която служи за избор на определени колони от дадено отношение.

## Q

**Query (заявка)** - логически израз върху обектите и взаимодействията, дефинирани в схемата. Резултат от изпълнението на дадена заявка е логическо подмножество на базата от данни.

**Query language (език за заявки)** – специализиран език за извличане на данни от базата. Вж. също **Data Manipulation Language, QBE, SQL**.

**Query-by-Example (QBE) (заявка-по-пример)** – непроцедурен визуален език за извличане на данни от релационни бази. Използува шаблони на заявки, които потребителят попълва с примери, описващи необходимия резултат.

**Querying (извличане на данни от базата)** – процесът на формулиране на заявка към база от данни с помощта на език за обработка на данните. Вж. също **Data Manipulation Language, Query language, SQL, QBE**.

## R

**Redundant Arrays of Independent Disks - RAID (масиви от независими дискове)** – хардуер, позволяващ ускорено извличане на мултимедийна информация.

**Record (запис)** - множество от свързани помежду си полета. В релационния модел съвпада с понятието кортеж. Вж. също **Tuple**.

**Recovery control (управление на възстановяването)** – процесът на възстановяване на данните в предишното им състояние от СУБД при хардуерни повреди и грешки на софтуера. Отменят се всички промени в базата, направени от изпълняващата се програма.

**Recursive closure (рекурсивна обвивка)** – операция, която не може да се зададе със средствата на релационната алгебра. Прилага се при наличието на рекурсивна връзка между кортежи от един и същ тип - например „служител - началник“.

**Recursive relationship (рекурсивна връзка)** – отразява взаимодействия между екземпляри на един и същ тип обект. Например типа връзка „НАЧАЛНИК – ПОДЧИНЕН“, която показва взаимодействието между служителя и прекия му ръководител. И двата екземпляра принадлежат на един и същ тип обект, в смисъл че и началникът, и подчиненият са служители на една и съща организация. За да се различават помежду си, на тях им се присвояват роли. Вж. също **Relationship type, Role**.

**Redundant data (повтарящи се данни)** – вж. **Data redundancy**.

**Referential integrity (ограничение за цялостност на връзките)** – задава се между две отношения и се използва за поддържането на непротиворечиви данни между кортежите им. Неформално подобно ограничение показва, че даден кортеж  $t_1$  от отношението  $r_1$  е свързан със съществуването на кортеж  $t_2$  в отношението  $r_2$ . Формално се изразява чрез понятието "чужд ключ". Вж. също **Foreign key**.

**Relation (отношение, релация)** - произволно подмножество на декартовото произведение на домейни. Неформално – таблица с редове и колони. Вж. също **Domain**.

**Relational database (релационна база от данни)** – множество от отношения. Вж. също **Relation, Relational model**.

**Relational DBMS (релационна СУБД)** – СУБД с релационен модел на данните. Данните се съхраняват в свързани помежду си таблици. Вж. също **Database Management System, Data model, Relational model**.

**Relational model (релационен модел)** – модел на данни, при който основна структура за представяне на данните в базата е отношението. Вж. също **Data model, Relation**.

**Relationship (връзка, взаимодействие)** – описва взаимоотношенията между обектите. По брой участващи в тях обекти връзките биват: рекурсивни, двоични, троични и пр., а по вид 1:1, 1:N, M:N.

**Relationship occurrence (състояние на връзката)** – отделен представител на даден тип връзка. Вж. също **Relationship type**.

**Relation schema (релационна схема)** – крайно множество от имена на атрибути. Вж. също **Schema, Attribute**.

**Relationship type (тип на връзката)** – произволно взаимоотношение, дефинирано върху типове обекти. Вж. също **Entity type**.

**Replication (дублиране)** - процес на определяне на копия (replica) на отделните фрагменти, които ще се съхраняват на различни възли.

**Report (отчет)** – оформено представяне на данни от базата. Повечето СУБД предоставят средства за тяхното проектиране и генериране. Вж. също **Database Management System**.

**Repository (хранилище)** – място където се съхраняват и поддържат данни.

**Requirements collection and analysis (събиране и анализ на изисквания)** - фаза на процеса „проектиране на базата“, при която се идентифицират и анализират потребителските намерения. В резултат се получава описание на изискванията към базата от данни.

**Role (роля)** – средство за маркиране на обекти от един и същи тип при участието им в рекурсивни връзки с цел тяхното отличаване. Вж. също **Recursive relationship**.

**Row (редица)** – вж. **Tuple**.

## S

**Schema (схема)** – описание на базата от данни. При релационните бази схемата дефинира таблиците и полетата във всяка от тях. Съхранява се в каталога. Вж. също **System catalog**.

**Second normal form (2NF) (втора нормална форма)** – при нея всеки атрибут на дадена таблица в първа нормална форма, който не е част от главния ключ, е пълно функционално зависим от него. Вж. също **Full functional dependency**.

**Secondary index (вторичен индекс)** – това е индекс, дефиниран върху произволно поле на файла с данни, по което се търси често. Вж. също **Index**.

**Security (сигурност)** – защита на базата от неправомерен достъп. Вж. също **Data security**.

**Select (избор, селекция)** - операция на релационната алгебра, която служи за избор подмножество от кортежи в дадено отношение, които удовлетворяват предварително

зададено условие (критерий) за търсене. Действието ѝ представлява по същество, филтриране на кортежите и извеждане само на онези от тях, които удовлетворяват зададеното условие.

**Sequential file (последователен файл)** - вж. **Heap file**.

**Server (сървър)** – софтуерно приложение, което осигурява определени услуги на клиенти. От гледна точка на технологията „бази от данни“ сървърът обработва заявки към базата. Вж. също **Client, Client-server**.

**Simple attribute (прост атрибут)** – атрибут, чиито стойности са атомарни (неделими).

**Single-valued attribute (едностойностен атрибут)** - атрибут, който приемат единствена стойност за даден обект.

**Snowflake schema (схема от тип “снежинка”)** - вариант на схемата от тип “звезда”, при който таблиците на измеренията са нормализирани. Вж. също **Star Schema**.

**Software engineers (софтуерни инженери)** – вж. **System analyst, Application programmer**.

**Specialization (специализация)** - процес на дефиниране на множество от подкласове за даден тип обект въз основа на специфични за тях характеристики.

**Spurious tuples (фалшиви кортежи)** - допълнителни кортежи при изпълнение на операцията „съединение”, които представят некоректни данни. Те се получават при лошо подбран свързващ атрибут, който не е нито главен, нито чужд ключ за някое от двете отношения. Вж. също **Join**.

**State (състояние)** – вж. **Instance**.

**Star Schema (схема от тип “звезда”)** - логическа структура, състояща се от таблица на фактите, заобиколена от таблици на измеренията, които могат да не бъдат нормализирани. Фактите са генерирани от събития, случили се в миналото. Таблицата на фактите може да само се чете и съдържа предимно цифрови данни. Таблиците на измеренията съдържат описателна текстова информация.

**Storage Area Network–SAN (локална мрежи за съхранение)** - високоскоростна мрежа, която предоставя периферни устройства за запазване на данни и при необходимост се прикрепя гъвкаво към сървърите.

**Structural constraints (структурни ограничения)** – това са ограничения върху типовете връзки. Биват два вида: кардинално число и ограничение за участие. Вж. също **Relationship type**.

**Structured query language (SQL) (структурен език за заявки)** – основен непроцедурен език за работа релационни бази. Използува фиксирана структура за описание на търсения резултат. Стандартизиран със стандарти SQL2 и SQL3, може да се вгражда в език за програмиране. Понастоящем почти всички релационни СУБД предлагат SQL, като реализациите се отличават по диалекта на езика.

**Superkey (суперключ)** –в релационния модел/ E-R модела това е съвкупност от атрибути, чиито стойности се използват за различаване на отделните кортежи/ обекти.

**System analyst (системен аналитик)** – хора, които определят изискванията на крайните потребители и разработват съответните спецификации на приложенията.

**System catalog (системен каталог)** – съдържа данни за структурата на базата, потребителите, приложенията, т.е. описание на всяка информационна единица, както и различни ограничения, наложени върху нея.

## T

**Table (таблица)** – съвпада с **Relation**.

**Ternary relationship (троична връзка)** – връзка между три типа обекти. Вж. също **Relationship type**.

**Testing (тестване)** - фаза от жизнения цикъл на приложение от тип “база от данни, при която се изпълнява приложението с цел намиране и отстраняване на грешки.

**Third generation language (3GL) (език от трето поколение)** – процедурен език за програмиране, при който се указва какви обработки да бъдат извършени и начинът, по който да стане това.

**Third normal form (3NF) (трета нормална форма)** - при нея всеки атрибут на дадена таблица във втора нормална форма, който не е част от главния ключ не зависи транзитивно от него. Вж. също **Transitive functional dependency**.

**Top-down approach (подход „отгоре-надолу“)** - подход за проектиране, при който схемата на конкретна релационна база от данни се получава в резултат от следните стъпки:

- проектиране на концептуална схема на базата от данни, като се използва модел от високо ниво (например E-R модел);
- изобразяване на получената схема в множество от отношения;
- анализ на отношенията с помощта на функционалните зависимости, като се прилагат четирите неформални мерки за качество;
- провеждане на нормализация до трета нормална форма, за да се избегнат транзитивните и частичните функционални зависимости.

**Transaction (транзакция)** – това е множество от операции върху обектите в схемата. Логически неделима единица работа към базата, която запазва нейната състоятелност.

**Transitive functional dependency (транзитивна функционална зависимост)** - функционална зависимост  $X \rightarrow Y$ , при която съществува подмножество от атрибути  $Z$ , което не е подмножество на главния ключ, и са в сила:  $X \rightarrow Z$  и  $Z \rightarrow Y$ . Вж. също **Functional dependency**.

**Transparency (прозрачност)** - означава свойството на системата да „скрива“ от потребителя начина на разпределение на данните и разположението им по отделните възли на мрежата.

**Tuning the database (настройване на базата)** - процес на промяна на първоначалния проект на физическата база (например добавяне или премахване на индекси) с оглед постигането на по-добра производителност на системата.

**Tuple (кортеж)** - произволен ред в отношението. Вж. също **Record**.

## U

**Union (обединение)** - множествена операция на релационната алгебра, приложима върху две съвместими по обединение отношения. Резултатът съдържа кортежите, които фигурират или в едната, или в другата, или и в двете участващи таблици, като

дублираните редове се елиминират. Еквивалентна е на операцията „обединение” на две множества. Вж. също **Union compatible**.

**Union compatible (съвместим по обединение)** – двете отношения, участващи в множествените операции „обединение”, „сечение” и „разлика” на релационната алгебра са с еднакъв брой атрибути, които по двойки приемат стойности от един и същ домейн. Това е необходимо, да се получава винаги правилно резултатно отношение. Вж. също **Union, Intersection, Difference**.

**Universe of Discourse (предметна област)** – частта от реалния свят, която се представя в базата. Съдържа информация за отделни единици от обкръжаващата ни действителност, за които се събират данни.

**Update (обновяване, актуализация)** - операция, която служи за променяне на стойностите на един или няколко атрибута в дадено отношение.

**Update anomalies (аномалии при обновяване)** – възникват при изпълнение на операциите Insert, Delete, Update поради излишество при съхраняване на данните в дадена таблица. В резултат се нарушава цялостността на данните. Вж. също **Data redundancy, Data integrity, Insert, Delete, Update**.

**User view (потребителска представа)** - разбирането за структурата и съдържанието на базата на даден потребител или приложение.

## V

**View (престава)** – таблица, която съдържа “виртуални данни”, извлечени от базата, без да се съхраняват явно в нея.

**View integration approach (подход „интегриране на представите”)** - подход, при който за всяка потребителска група се проектира представа съгласно нейните изисквания. След това проектираните схеми се интегрират в концептуална схема за цялата база.

## W

**Weak entity (слаб тип обект)** - тип обект, който не притежават ключов атрибут. Отделните екземпляри се идентифицират чрез свързването им с екземплярите на друг тип обект. Вж. също **Entity type, Key**.

X

Y

Z