**Model de date**

-Descrierea modului in care datele sunt reprezentate

-Implica identificare struct de date, a operatorilor si a restrictiilor de integritate

**Structura de date**:

-Modalitate de stocare a informatiilor intr-un sist inf pt a le utiliza eficient

-col de date au o serie de relatii care conduc la un anumit mecanisc de selectie si identificare (Relatii: 1-1,1-M,M-M)

-**Clasificare**: acces secvential/direct, omogene/eterogene, statice/dinamice, logice/fizice

-In functie de tipurile de legaturi:

-punctuala

-liniara

-arborescenta

-retea

-relationala

##

**Operatori Algebra relationala(univ,baza,derivati)**

-Reuniunea, Diferenta, Produs Cartezian

-Proiectia, Selectia

-Jonctiunea -caz particular de produs cartezian(Interna - egalitate, naturala, inegalitate, Externa - La stanga,dreapta,completa)

-Intersectia, Diviziunea

-Complementarea,spargerea unei relatii, inchiderea tranzitiva (Op extinsi)

##

**Operatori din calculul relational**

-Conectivele: conjunctia, disjunctia, negatia

-Cuantificatorii: existential, universal

**Restrictii de integritate**

-S.n si constrangeri si definesc cerintele pe care trebuie sa le satisfaca datele

-Rolul lor e de a pastra datele corecte, consistente si coerente in procesul de culegere, stocare, prelucrare, transmitere, extragere

-Se impart in : **Restrictii structurale** ( unicitatea cheii (PK), referentiala (FK), entitatii (Not NULL) si r**estrictii de comportament:** UNIQUE, CHECK

##

**Tipologia modelelor de date:**

-Componente: structura de date (LDD), operatorii (LMD) si restrictii de integritate (LDD si nu numai)

-**Categorii**: model conceptual, logic, fizic

-Dupa modul de definire: arborescent, retea, relational, orientat-obiect

##

**Baze de date**

**-**Un ansamblu de colectii de date aflate in interdependenta, impreuna cu dictionarul de date care contine descrierea datelor si a legaturilor dintre ele, asigurand interdependenta datelor fata de aplicatii

-Dpdv al **modului de organizare**: ierarhice, retea, relationale, orientate obiect, NOSQL

-Al **domeniului de aplicare**: universale, specializate

-Dupa **gradul de centralizare**: centralizate, distribuite. **Caracteristici. niveluri:**

-conceptual (viziunea proiectantului, schema conceptuala)

-logic (viziunea dezvoltator, schema logica), -fizic(viziunea inginerului de sistem, schema fizica

-Strucutrata conform unui model logic de date

-Redundanta minima

-Accesibila mai multor utilizatori in timp util

**Etape de realizare:**

**-**Analizarea cerintelor

-Proiectarea bazei de date

-Implementarea bazei de date

-Punerea in functiune a BD

-Mentenanta BD

##

**Sisteme de baze de date**

-un ansamblu de elemente interconfitionate care contribuie la realizarea si exploatarea unei app cu BD

-**componente**: date, software, elemente aux

-**Arhitecturi:**

-reprezentare grafica a elementelor sistemului si a legaturilor, evidentiindu-se legatura cu exteriorul

-arhitectura de componente

-arhitectura pe niveluri de organizare a datelor

##

**Sisteme de gestiune a bazelor de date**

-un ansamblu complex de programe care asigura interfata intre o BD si utilizatori, are rol de a gestiona BD

-Partea principala a componentei software a unui SBD, care interactioneaza cu toate celelalte componente si cu exteriorul

-**Obiective** (Vizeaza capacit de a furniza software complet pt dezv de apl cu BD) (**Generale**)

(**Spepcifice**) : independenta fizica si logica

-redundanta minima si controlata

-facilitati de utilizare a datelor (SQL)

-protectia datelor (securitate, integritate)

-Partajabilitatea datelor

-legaturi dintre date

-performante globale

##

**Functii ale SGBD**

1. Descrierea datelor (LDD)

-LDD conceptual: Create, Alter, Drop

-Ldd logic: Create view

-LDD fizic: Create Cluster/Index

2. Manipularea datelor (LMD) - cea mai complexa functie utilizata in SGBD

-operatii de incarcare si validare

-actualizare: insert,update,merge, del

-prelucrare a datelor

-interogare a datelor

3.Utilizare: toate interfetele de dezvoltare a aplicatiilor cu BD

4. Administrare: totalitatea instrumentelor de adminsitrare

**Tipolgie:**

**-**Dupa limbaj: SGBD cu limbaj gazda/propriu

-Dupa mode logic: iererhice, retea, relationale, OOP

-Dupa localizrea BD: centralizate, distribuite

##

**Modelul relational**

-Alcatuit din structura relationala, operatori relationali, restrictii de integritate

**Structura:**

-Domeniu (nume)

-Relatie

-Atribut

-Tuplu

-Cheie

-Schema

-Operatori relationali (algebra relationala)

-Restrictii de integritate

##

**Normalizarea datelor**

-Se efectueaza in cadrul etapei de proiectare a BD, cu scopul de a optimiza schema conceptuala prin:

-Eliminarea animaliilor de actualizare (stergere, adaugare, modificare)

-Inlaturarea redundantei datelor

-Trecerea unei relatii dintr-o forma normala in alta presupune eliminarea unui anumit tip de dependente (functional partiale, traanzitive, multivalorea,jonctiune), dependente care determina aparitia anomaliilor

##

**FORMA NORMALA 1**

-BDR se afla in FN1 daca toate relatiile componente sunt in FN1

-O relatie este in FN1 daca **valorile** asociate atributelor se afla la **nivel elementar** si daca **nu exista atribute generatoare de valori repetitive**

-Presupune eliminarea atributelor compuse si a celor repetitive

##

**FORMA NORMALA 2**

-BDR se afla in FN2 daca toate relatiile componente sunt in FN2

-O relatie este in FN2 daca este in FN1 si **oricare din atributele non-cheie este** **dependent functional complet de atributele care formeaza cheia primara a relatiei**

-FN2 interzice manifestarea dependentelor functionale partiale

-Presupune eliminarea dependentelor funct part din FN1 prin :

-crearea unei noi relatii, eliminandu-se atributul care formeaza determinantul dependentei

-Determinarea cheii primare a fiecarei noi relatii ca fiind formata din atributele care reprez determinantul dependentei functional partiale

##

**FORMA NORMALA 3**

-BDR se afla in FN3 daca toate sunt in FN3

-O rel este in FN3 daca este in FN2 si **atributele non cheie nu sunt dependente tranzitiv de cheia primara**

-FN3 interzice manifestarea dependintelor tranzitive in cadrul relatiei

-Se realizeaza prin eliminarea dependentelor funct tranzitive prin crearea unor noi relatii

**BNCF-** o relatie se afla in BNCF daca dep funct care se manifesta au in partea stanga o cheie candidata. Se aduce in BCNF prin eliminarea tuturor tipurilor de dependente non-cheie

##

**SQL (Structured Query Language): 1. LDD**

-Nivel conceptual: CREATE/DROM/ALTER DB/TABLE

-Nivel logic: CREATE/DROP VIEW

-Nivel fizic: CREATE/DROP INDEX/CLUSTER

**Modificarea structurii unei tabele:**

-Adaugare coloane **ADD**

-Modificare coloane : **Modify**

-Specificarea unor restrictii: **ADD/DROP/EN/DIS**

- redenumirea unei coloane: **RENAME**

**-PK COMPUSA este o cheie formata din 2 sau mai multe atribute care joaca rol de cheie primara**

- PK compusa se face doar la nivel de tabela

-NOT NULL se face la niv de coloane

-FK se sterge prin : ON DELETE CASCADE, ON DELETE SET NULL

-Crearea unei tabele prin copierea din alta tabela nu ia decat NOT NULL

-Se poate recupera tabela stearga prin **FLASHBACK**

**##**

**SQL: 2. LMD**

-Comenzi : SELECT, INSERT, UPDATE, MERGE, DELETE

-Clauze SELECT: WHERE, GROUP BY, HAVING, ORDER BY

-Utilizarea expresiilor poate fi facuta in:

-Lista de coloane din SELECT

-In WHERE, HAVING, ORDER BY

-in VALUES din cadrul INSERT

-In SET si WHERE in cadrul UPDATE

-In WHERE din cadrul DELETE

##

**Functii SQL**

-UPPER/LOWER

-INITCAP

-CONCAT

-LPAD/RPAD(s1,s2,n)- Umple stringul s1 cu s2 pana la N caractere

-LTRIM/RTRIM(s1,s2)- Scoate toate caracterele din s2 din stanga/dreapta

-TRIM - scoate caractere din stanga/dreapta

-LENGTH

-INSTR- returneaza pozitia unui subsir

-SUBSTR

-REPLACE - inlocuieste un subsir

-ROUND(n,i) - rotunjeste un numar la i zec

-TRUNC - trunchiaza un numar la i zec

-MOD

-SYSDATE

-CURRENT\_DATE

-ROUND

-TRUNC

-NEXT\_DATE

-LAST\_DAY

-ADD\_MONTHS

-MONTHS\_BETWEEN

-EXTRACT

-NVL(e1,e2) - returneaza e2 daca e1=null

-NULLIF(e1,e2) - returneaza NULL daca ei=e2, altfel returneaza r1

-DECODE - verifica variante alternative

##

**SQL - Gruparea datelor**

Clauze in SELECT

-**GROUP\_BY** grupeaza liniile in scopul identificarii valorilor comune grupurilor

-**HAVING** restrictioneaza grupurile create prin clauza GROUP\_BY pe baza unei conditii asupra functiilor de grup

-Agregarea datelor

-**ROLLUP**: creeaza subtotaluri

-**CUBE**: creeaza subtotaturi pentru fiecare combinatie posibila de dimensiuni

-**GROUPING** **SET**: defineste grupuri multiple intr-o singura cerere

-**GROUPING**: identifica liniile agregate prin utilizarea ROLLUP sau CUBE

##

**SQL - Subcereri**

**Subcerere** = comanda SELECT inclusa intr-o alta comanda SQL care poate returna una sau mai multe linii

**Utilizare**: pt a obtine rezultate pe baza unor criterii

-a crea tabele pe baza unora deja existente

-a manipula seturi mari de date

-a crea tabele virtuale

-a defini vederi dinamice

-a defini expresii calculate dinamic prin subcereri scalare

**Tipuri** - single-row

-multiple-row

multiple-column: mai mult de o coloana

-corelate

-scalare

**Operatori de comparatie:**

-IN

-NOT

-ANY compara valoarea cu oricare valoare returnata

-SOME

-ALL compara valoarea cu toate valorile returnate