

Database: a collection of related data.

Database Management System(DBMS):

a software system to facilitate the creation and maintenance of a computerized database.

Database System: Database+DBMS+Application Programs:-

the DBMS software together with the data itself. Sometimes, the applications are also included(software+database).

Advantages of DB system:

- * Redundancy can be reduced (البيانات في مكان واحد واكثر من يوزر يقدر يوصل ليها على عكس ملفات الاكسل).
- * Restricting Unauthorized Access (يقوم بإنشاء اليوزرز واعطاء الصلاحيات لهم على البيانات) .
- * Enforcing Integrity Constraints (بطبق قواعد معينة على البيانات - اني مثلا مسمحش بتكرار رقم البطاقة لاكثر من موظف).
- * Inconsistency can be avoided (الداتا في مكان واحد ولما بيحصل ابديت الكل بيشوفه في نفس الوقت على عكس تخزينه) .

Disadvantages of Database

- It needs expertise to use (يحتاج أعين شخص يعرف يستخدمها)
- DBMS itself is expensive
- The DBMS may be incompatible with any other available DBMS

Database Users:

طبيعة استخدام الشركة للداتا بيز وكم يوزر هيتعامل معها وسرعة(Step 1:- Analysis and requirements gathering (نوم وزيادة البيانات قد ايه).

Step 2:- Database Design.

(بحول الديزائن لداول وداتا بيز او بكت واجهز السوفتوير ككل) .

Step 4:- Application development (اليوزر انترفيسيس اللي المستخدم هيتعامل معها).

Responsibilities for each step:

Step 1:- System analysts.

Step 2:- Database Designer -create db design- (conceptual schema).

Step 3:- DBA (Database Administrator) -install DBMS - Create tables on the DBMS – create users and authorize access to db – maintain db performance- .

Step 4:- Application programmer.

Three Schema Architecture:

اللي بيشوفه المستخدم وازاي بتظهر ليه:- External Schema:-

Conceptual Schema:- (logical model) يحتوي على كل الجداول وال العلاقات بين الداتا.

Physical Schema:- (physical model) ازاي الداتا متخزنة على الديسك بناعي و مكانها فين و المساحة المتاحة المشغولة .

Data independence: لما أعمل تغير في سكيماء معينة مش شرط السكيماء الأعلى تحس بالتغيير ده .

Mappings: it is the processes of transforming requests and results between levels(schema).

DBMS functions:

- Multimedia Function
- Spatial Data
- Time Series
- Data mining

Database Environment:

- Centralized database (single point of failure)
 - Distributed database →(no single point of failure)
- 1-Replication(partial – full)
بنسخ الداتا بيز
2-Fragmentation .
بقسم الداتا بيز .

*****start of conceptual design based on customer's requirements*****

Entity Relationship Modeling(ERD): طريقة يستخدمها علشان اعمل الكومنسيشنال ديزاين .-
يتعرف على طلبات العميل بناعي وبعرضها في شكل كيانات و علاقات بين الكيانات وبعضاها

Entity: كل اوبجكت او شيء مستقل بذاته موجود على السيستم بناعي زي عميل في بنك .

Attribute: . الصفات اللي بتوصف الكيان .

*P.S. Attribute = Columns.

1- **Single/Simple Attribute:** . بيحمل قيمة واحدة يعني الموظف ليه رقم بطاقة واحد .

2- **Multi-valued Attribute:** . بيتحمل اكتر من قيمة زي رقم التليفون .

3- **Composite Attribute:** هو اtribيوت بينقسم لاكثر من جزء زي اني أقسم العنوان لشارع ومنطقة .

4- **Derived Attribute:** هو اtribيوت مبني على حسابات زي حساب العمر من تاريخ الميلاد .

Candidate key: لما كيان يكون عنده اtribيوت او اكتر يوينيك

Strong Entity: كيان بيكون عنده اtribيوت ينفع يكون يوينيك (مميز)

Weak Entity: كيان معندوش يوينيك اtribيوت ووجود الكيان معتمد كليا على كيان آخر

Identifying Relationship: Between strong entity and weak entity.

Relationship: connection between entities.

1-Degree of a relationship: the number of participating entities (Unary-Binary-Ternary). (عدد الكيانات المشاركة في العلاقة)

***Binary Relationship:** between two entities.

***Unary/Recursive Relationship:** between the entity and itself.

***Ternary Relationship:** between three entities.

2-Cardinality ratio: specifies the maximum number of relationship (كل كيان يشارك بایه) (one to one – one to many – many to many).

***1:1 (one to one)**

Example:

- The relation between car and employee.
- An employee uses at most one car.
- A car is used by at most one employee.

***1:M (one to many)**

Example:

- An employee uses at most one car.
- A car can be used by several (many) employees.

***M:M (many to many)**

Example:

- An employee uses several (many) cars.
- A car can be used by several (many) employees.

3-Participation: specifies the minimum number of relationship (must-may).

*****end of conceptual design based on customer's requirements*****

*****start of logical design transforming it from conceptual design*****

tuple = row

نقطاع العمود مع الصف ويجب ان يكون قيمة واحدة فقط = domain

Step 1: mapping regular entity (تحول كل كيان لجدول من صفات وأعمدة وقيود).

كل اtribut بيتتحول لعمود في الجدول*

*composite attribute: .

*multi-valued attribute: . يعمله جدول منفصل وبتحط فيه فورين كي: .

*derived attribute: . مش بيظهر معاليا لأنه بيقوم على العمليات الحسابية وبيبطئ الأداء: .

Step 2: mapping weak entity

(يعمل فورين كي يقابل البرميري كي في الكيان الأب)

Step 3: mapping relationships

**binary/unary relationship (one:many) : add pk from one side table as fk to Many side table (مثلاً بالخذ من جدول القسم البرميري كي بتاتعه وبتحطه لك فورين كي في جدول الموظفين وهبقي قدام الموظف رقم القسم بتاتعه)
بالخذ البرميري كي وبتحطه في نفس الجدول لك فورين كي: *unary:

**binary/unary relationship (many:many) : add PKs from parent tables as fk to a new table (يعمل جدول جديد وبالخذ البرميري كي للجداولين في العلاقة وبتحطهم لك فورين كي في الجدول الجديد)

**binary/unary relationship (one:one) :

. بالخذ البرميري كي من الجدول الماي وبتحطه لك فورين كي في جدول الماست: 1- may-must:

. بالخذ البرميري كي لأي جدول وبتحطه في الثاني لك فورين كي 2- may-may:

3- must-must: بدمج الجدولين مع بعض .

**ternary relationship: add PKs from parent tables as fk to a new table
(عمل جدول جديد وبأخذ البريمري كي للجداول في العلاقة وبخطهم ك فوريين كي في الجدول الجديد)

*****end of logical design transforming it from conceptual design*****

*****start of physical design*****

SQL → DDL(data definition language) – DML(data manipulation language) – DCL(data control language).

DDL(data definition language): manipulate structure of db (مسؤوله عن الستراتيجي ببنية البيانات)
(Create – edit – delete).

-**CREATE:** CREATE TABLE table_name (cl_name CHAR(10) KEY, cl_name CHAR).

-**ALTER:** ALTER TABLE table_name ADD cl_name./ALTER TABLE table_name DROP cl_name/
ALTER TABLE table_name ALTER COLUMN col_name CHAR(50).

-**DROP:** DROP TABLE table_name.

-**TRUNCATE:** TRUNCATE TABLE table_name.

DCL(data control language): use commands to give privilege access to data
(إعطاء صلاحيات للمستخدم على حاجات معينة في البيانات)

-**GRANT:** GRANT SELECT ON TABLE employee TO AHMED

*GRANT ~~ ON TABLE ~~ TO ~~ WITH GRANT OPTION (يعني اليوزر يقدر يدي الاوشن ده لغيره)

-**REVOKE:** REVOKE SELECT ON TABLE employee FROM AHMED.

DML(data manipulation language): manipulate the data

-**INSERT:** INSERT INTO table_name (col_name,col_name) VALUES (**,**).

-**UPDATE:** UPDATE table_name SET col_name WHERE col_name=**.

-**DELETE:** DELETE FROM table_name WHERE cl_name=**.

-**SELECT:** SELECT * FROM table_name WHERE cl_name=**.

DELETE: delete all data but keeps the structure, can include a where clause, **can be rolled back**, keeps the physical memory assigned to data until a commit is issued.

TRUNCATE: delete all data **but keeps the structure**, **cannot** include a where clause, **cannot** be rolled back (auto commit), **de-allocates** the physical memory assigned to data.

DROP: delete all data and structure(table).

Database Schema: is a group of related objects in a database, has one owner who has access to manipulate the structure of objects in the schema.

Constraint: restrictions on database table to maintain integrity of data.

Primary Key: must contain unique value – cannot contain null value.

Not Null: cannot contain null value.

Unique Key: must contain unique value – can contain one null value.

Foreign Key: take primary key from table A to table B as FK. (PK as Parent record, FK as Child)

Check Key: customize specific rule(salary for employee between 100 and 200).

Normalization: عملية بفك فيها الجداول لجداول أصغر شوية لتحسين الديزайн

process of decomposing bad relations(tables) by breaking up their attributes into smaller relations.

To avoid:

- redundancy
- insert anomalies
- update anomalies
- delete anomalies

Functional dependency: كونسٹرینت بين عمودين:

Value of A uniquely determines the value of B.

مثلاً من خلال البريميри كي "رقم الهوية" اقدر اجيب اسم الموظف

Fully Dependent: عمود معتمد كلياً على البريميри كي

Partially: (non-key attribute depends on part of key attribute(composite pk)).

*composite pk: بريميри كي مكون من صفين او اكثر

مثلاً البريميри كي مكون من عمودين وفيه عواميد معتمدة على عمود واحد فقط منه.

Transitive: non-key column dependent on non-key column that dependent on key column

عمود أ مش بريميри كي معتمد على عمود ب مش بريميри كي وال ب معتمد على بريميри كي

Normal Form: condition using keys and functional dependency of a relation(table) to determine in which schema it is in → 1NF-2NF-3NF.

First normal form (1NF):

doesn't contain → Multi-valued Attribute – Repeating group – Composite attribute.

Repeating group: is multi-valued attributes relating to each other.

1-Take multi-valued columns and repeating group and pk as fk to a new table.

بأخذ العمود اللي فيه قيمتين والعواميد اللي ليها علاقة بعض والبريميри كي وبحطهم في جدول جديد.

2-Separate composite attribute into two columns in the same table.

Second normal form(2NF):

contain → 1NF,

doesn't contain → Partial dependency(بنظره في الجدول اللي فيه كومبزيت بريمرى كي).

Take the non-key attribute and the key that depends on and place them in a new table.

بأخذ العمود اللي معتمد على البريمري كي وبخطهم في جدول جديد هو والبريمري كي.

مثلاً جدول من 3 عواميد (أ، ب، ج) والبريمري كي هو أ، ب والج معتمد على ب فقط، اذن باخذ ج، ب وبخطهم في (جدول منفصل)

Third normal form(3NF):

contain → 2NF,

doesn't contain → Transitive dependency(non-key dependent on non-key dependent on key).

يبكون عندي عمودين مش بريمرى كي (أ، ب) أ معتمد على ب و ب معتمد على البريمري كي
باخذ أ، ب وبخطهم في جدول جديد وبخلي ب هو الفورين كي (لأنه هو المعتمد على البريمري كي في الجدول الأول)