Data_Base systems

Chapter 1:-

* Before Database we had filebase system

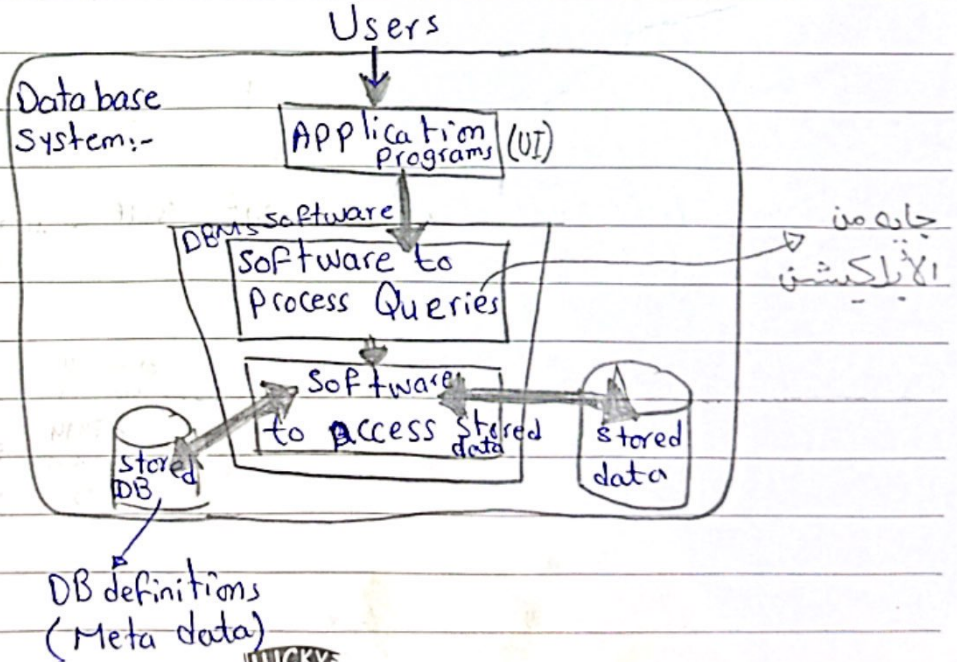
* filebase system has limitations
  1- separation and isolation of data (Excel, word)
  2- Duplication of data
  3- program Data Dependence (depend on one program)
  4- Incompatible file formats

* Database :- A collection of related data

* Database management system:- (DBMS)
  A soft ware package / system to facilitate the creation and maintenance of a computerized database

Data base + DBMS = Database System

⇒ Database Components:-



Meta data is data base definition stored all information

LUCKY

الداتا موجوده في مكان واحد

Database system advantages:- means

1- Controlling Redundancy (زيادة عن الحاجة)
2- Restricting Unauthorized Access ممنوع دخول لـ users معينين
3- sharing data
4- Enforcing Integrity Constraints (apply some rules on data)
5- Inconsistency can be avoided تناقضات
6- providing backup and Recovery

Disadvantages of data base system:-

1- Need expertise to use
2- DBMS is expensive
3- May be incompatible with any other available DBMS
   to solve it we can use 3rd party tool to simplify the transfer of data

DBA

- Create Users and authorize access to db
- Maintain DB performance

Data base Users

step 1:- Analysis and requirement gathering (مستخدمين، كل يوم؛ حجم ومعدل زيادة البيانات ... الخ)
budget + infrastructure

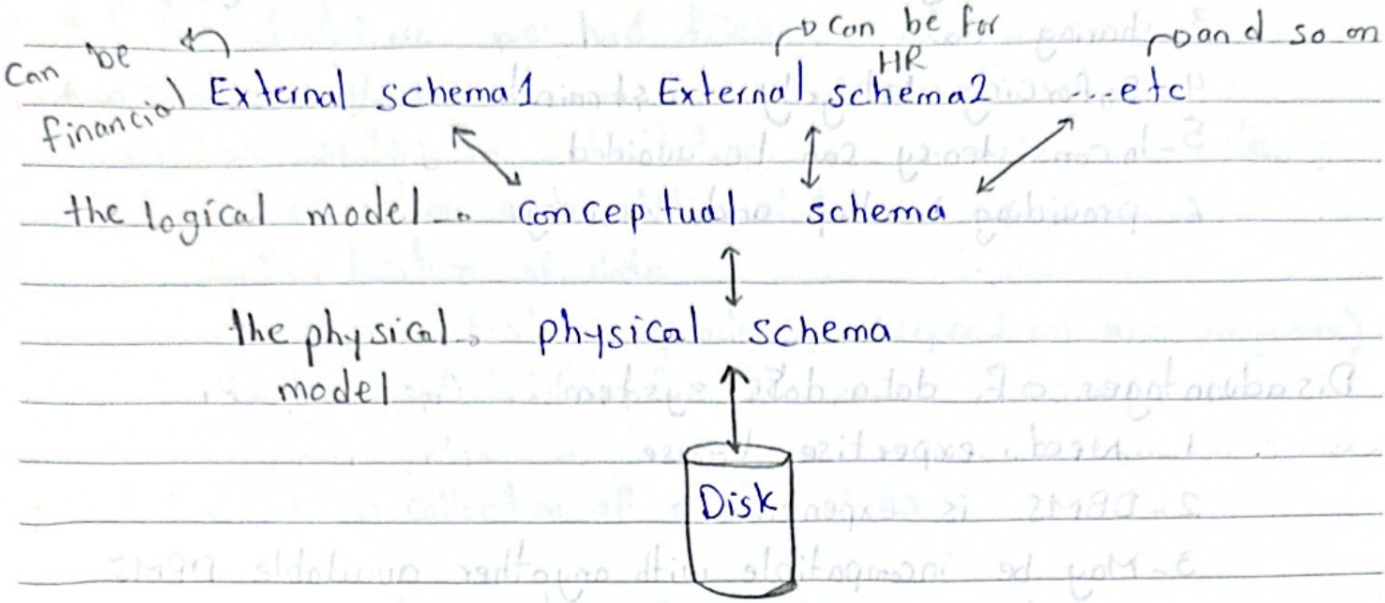
step 2:- Database design (How to convert it to classes ... etc)

step 3:- Implementation (setup data base managements & Infrastructure)

step 4:- Application development (user interface)

step 1	step 2	step 3	step 4	
* system Analyst	* Database designer	(DBA) Database Administrator	Application programmer	End User
• Business analysis and requirements gathering	create data base design (conseptual schema)	• Install DBMS • create DB schema and populate	• Develop & test & debug the application	

DBMS Architecture
(3 schema architecture)



① External schema:- is what user sees [Can Be more than one]
they are concerned with what data the user will see and how the data will be presented to the user

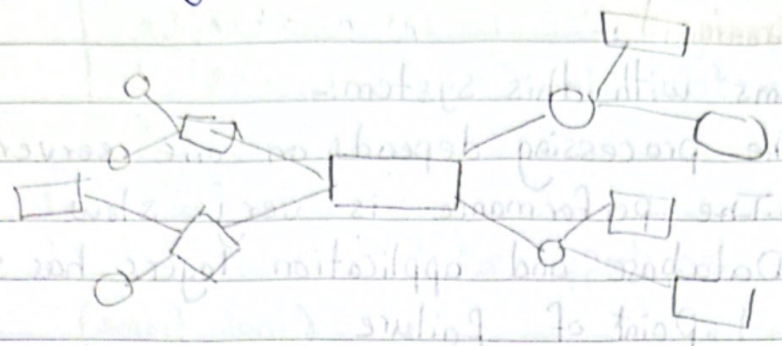
② Conceptual schema:- they are concerned with what is represented (define database structures such as tables and constraints)
→ logical model

③ physical schema:- How the data are represented in the database? How the data structures are implemented?
→ physical model

• Note :- they all are separated to create data independance that higher level architecture doesnot need to feel the changes in the Lower Level

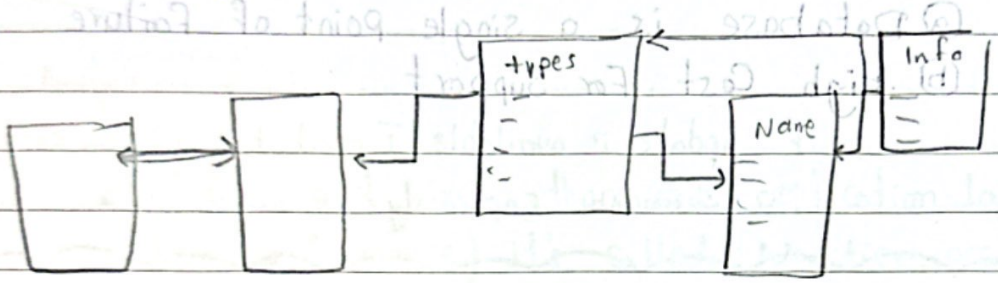
Data Models :-

- the logical model / conceptual model :-



provide concepts that are close to the way many users perceive data, entities, attributes and relationships

- physical data model :-



describes how data is stored in the computer and the access path needed to access and search for data

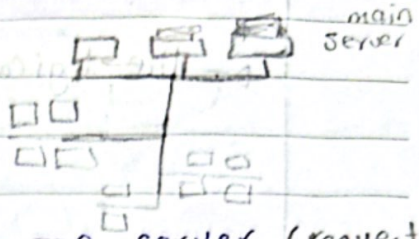
Mappings :- It is the processes of transforming requests and results between levels

DBMS and other functions :-

- text / Number / Image / Audio / Video (old / New)
- Spatial Data (GIS) geografical Information system support spatial data
- Time Series (sort verses time) I can take snap shot for data (لقطة)
- Data mining (clustering, classification, Association Rule
  - (Example E-commerce, Super market)
  - Can be built in but with minimum customization

LUCKY

Centralized database Environment



① Main frame Environment :-

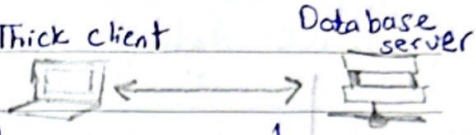
↳ problems with this system:-

a- The processing depends on one server (request o

b- The performance is very slow

c- Database and application layers has single point of failure (main frame)

② Client / Server environment :-



↳ problems with this system environment :-

ⓐ Database is a single point of failure

ⓑ High Cost For Support

↳ if update is available I need to update each machine Separetely

* called thick client because Application is locally setup and installed for each end user

↳ Advantages :-

@ Application layer isn't a single point of failure

③ Internet Computing environment (Three-tier architecture)

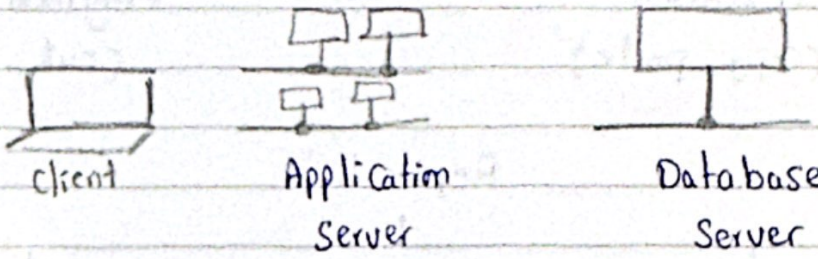
↳ Problems with this environment:-

a- Application server is a single point of failure

b- database is a single point of failure

↳ Advantages:-

a- Lower cost for support and maintenance



Note:-

- If we add more than one application layer (to improve the system performance) it's called N- tier archietecture

- for each application server number of users are connected

LUCKY

Distributed database

* it supports high availability of data base

it has 2 methods:-

- Replication (copy-paste)
- Fragmentation (cut-paste)

Replication:

- partial Replication: copy just a part of the system not all of it
- Full Replication: copy the whole server so that if the main stoped the copy work

Fragmentation

* distribute the system to fragments of data

- Horizontal
- Hybrid
- vertical columns

* each fragment has a server for it if any of them dis connected the system will maintain

Advantages:- 1. Database is Not a single point of failure

Disaduantages:- High cost

بشوف محتاجه بيقى upon running وعادى مش متأثر أوى واستحمل Centeralized without failure or down time

either Replication or Fragmentation → both will need setup and installation of DBMS software with license

- Same Vendor → Homogeneous Database
- Different Vendor → Heterogeneous Database

Chapter 02

Entity Relationship modeling:- (ERD)

Identifies information Required by the business by displaying the relevant entities and relationships between them

* what is entity?
  ↳ the entity is a thing in the real world with an independent exitance physical existence or conceptual existance

* each entity has characteristics (Attributes) ↓
  Example :-
  Entity :- student
  Attributes :- Name, age, ... etc

* Relationships is relation between different entities.

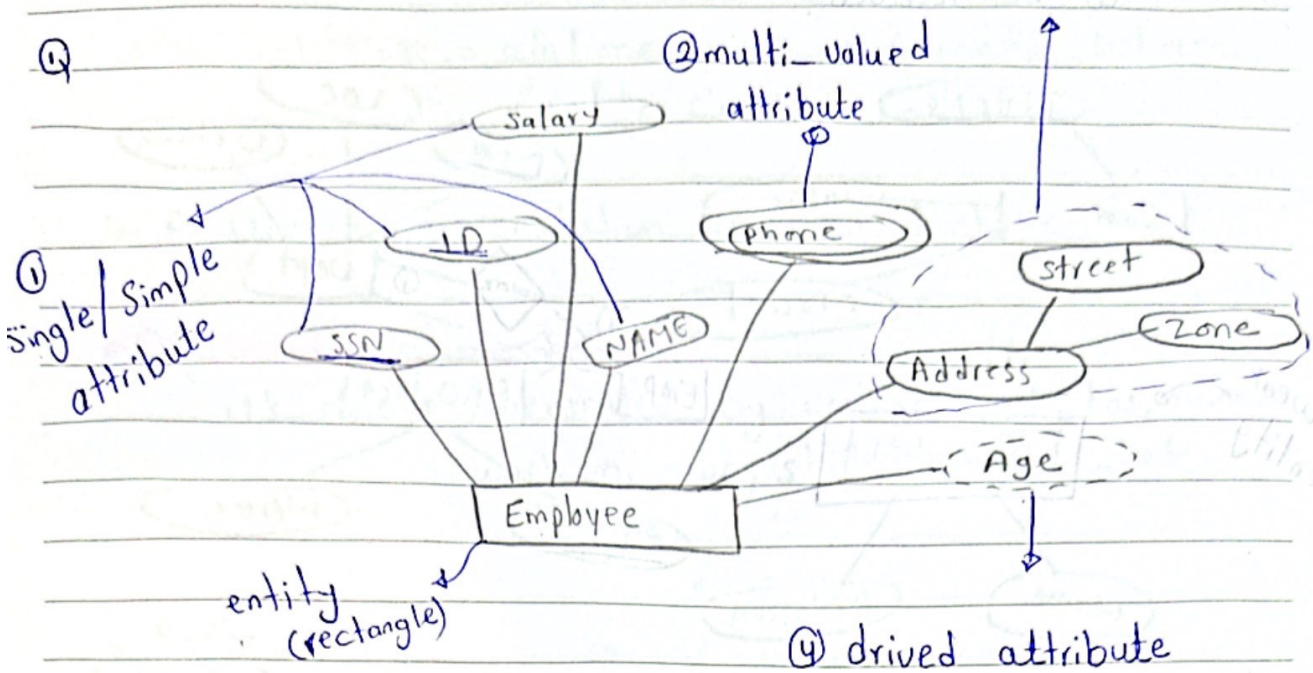
* In building a data model a number of questions must be addressed :-

① what entities need to be described in the model?
② what characteristics or attributes of these entities need to be recorded?

③ Can an attribute or a set of attributes be identified that will unigely identify one specific occurrence of an entity?

④ what associations or relationships exist between entities?

Types of Attributes:-



① **Single / simple attribute:-** Attribute that are not divisible and have a single value for a particular entity

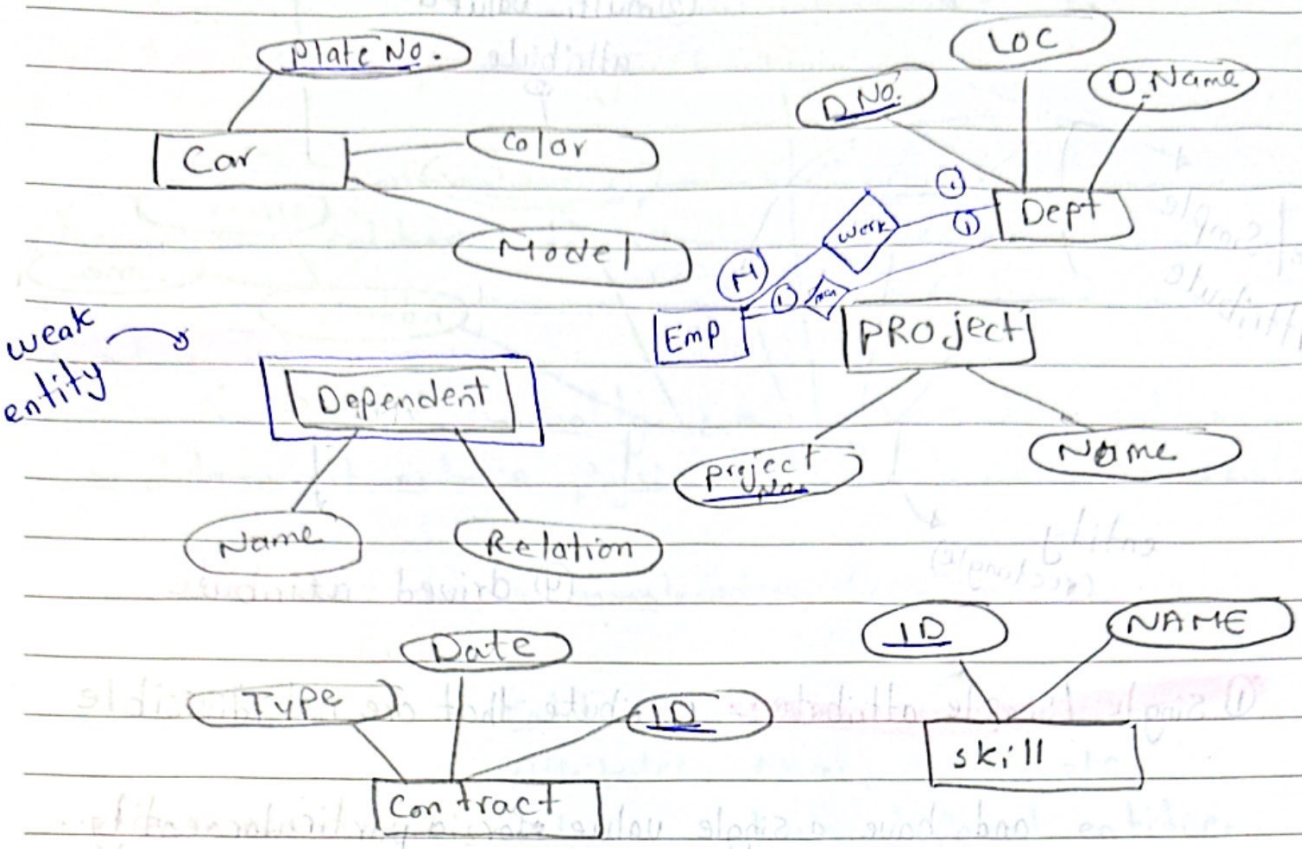
② **multi valued attribute:-** has a set of values for the same entity (phone)

③ **Composite attribute:-** Can be divided into smaller sub parts

④ **drived attribute:-** Can be calculated from another attribute or entity

* To define a variable as unique (special), a line is placed under it ex (ID)

2- what characteristics or attributes of those entities need to be recorded?



3- Can any attribute or a set of attributes be identified that will uniquely identify one specific occurance of an entity?

1- we put ___ to set that attribute is unique for the entity called Candidate keys.

2- when an entity type has more than one key, those are candidate keys

3- entities
- strong entity:- has unique identifier
- weak entity تحقق شرطين

① an entity that doesn't have a key attribute

② must be fully dependent on another entity.
employee على ال dependent

LUCKY

Relationships → Degree, Cardinality Ratio, participation

Q4

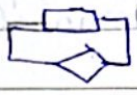
Relationships: a relationship is a connection between entity classes

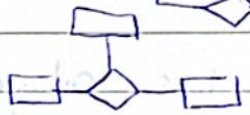
* Relation ships notation is dimond shape 

* **Degree of relation ship:-**
it's always represented by verb ⇒ work, has, manage, work on, own, skilled used ... etc

* **Degree:-**

① Binary Relationship:-  (one to one)

② Unary / Recursive relationship:-  (بين الحاجه وتقيها)

③ ternary relationship:-  (Join 3 entities)

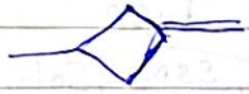
Cardinality Ratio:-

* it specifies the max number of relationships.

① one Ⓜ many  Ⓜ many to many

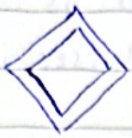
* Ternary relationship needs to be the same (many or one) if not break it to more than one relation 

Participation x Cardinality

- Specifies minimum number of relationship that each entity can participate with.
- Represented By:- ═══ Must
  ─── May 

LUCKY لو واحد بس هيكسر الـ Rule

Identifying relationship:- it's a relationship between the owner and weak entity

notation 

* I can add an attribute on a relationship 

*participation between weak entity and owner entity must be must

Chapter 3

Mapping → Converting Conceptual to logical

Relation data base:-

*each table consists of number of tuples (rows) and columns → the intersection between tuples and columns are called Domain (cell) and it has single value only

* the table has column or number of columns which are called primary key.

* the primary key has some characteristics :-
  1. it must contain an unique value for each raw of data
  2. it can not contain null values

To convert Conceptual design to logical design we have to follow some rules:-

1. Mapping of regular entity types

create table for each entity type

Emp (ID, SSN, Salary, Street, Zone)

Single attribute

Compound

* Primary key should be or prefered to be low size.

* to put multi value attribute :-

Emp (ID, SSN, Salary, Name, DoB, Street, Zone)
Emp – phone (SSN, Phone)
↳ foreign key (primary key as foreign key)
↳ Both of them the combination should be unique

* Drieved attribute better not to put it but if I use it always I should put it in the database

---

[2] step 2 :- Mapping weak entity

I take primary key of owner entity and add it as foreign key

Dependent (SSN, Name, Relation)

---

[3] step 3 :- Mapping of relationship types :- (Binary/Unary 1:N)

I add the primary key of the one side to many side

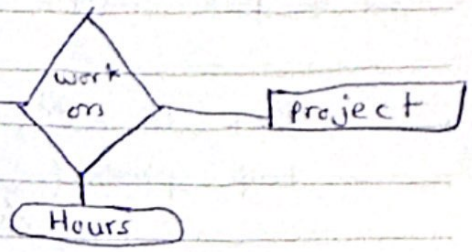
Fo

Lucky

step 4 : Mapping of relationship types :- M:N

we add it as new table

work_on (SSN, PNo, Hours)



Add Foriegn keys to the new table for both parent tables.

Step 5 : Mapping of relationship types 1:1

① may _ must :-
I take may side Primary key as foriegn key in must side

it has cases :-
1. may — may
2. may — must
3. must — must

② may _ may :- (ex :- own)

take 1st as foriegn in the second

take 2nd as foriegn in the first (Recommended)

create new table with Primary keys as foriegn

3 must _ must :- (ex :- Has)

* merge 2 tables as one table => delete Contract and add it to Emp

* if I need to use Contract ID later on as foriegn key I will use the primary key of Emp => SSN

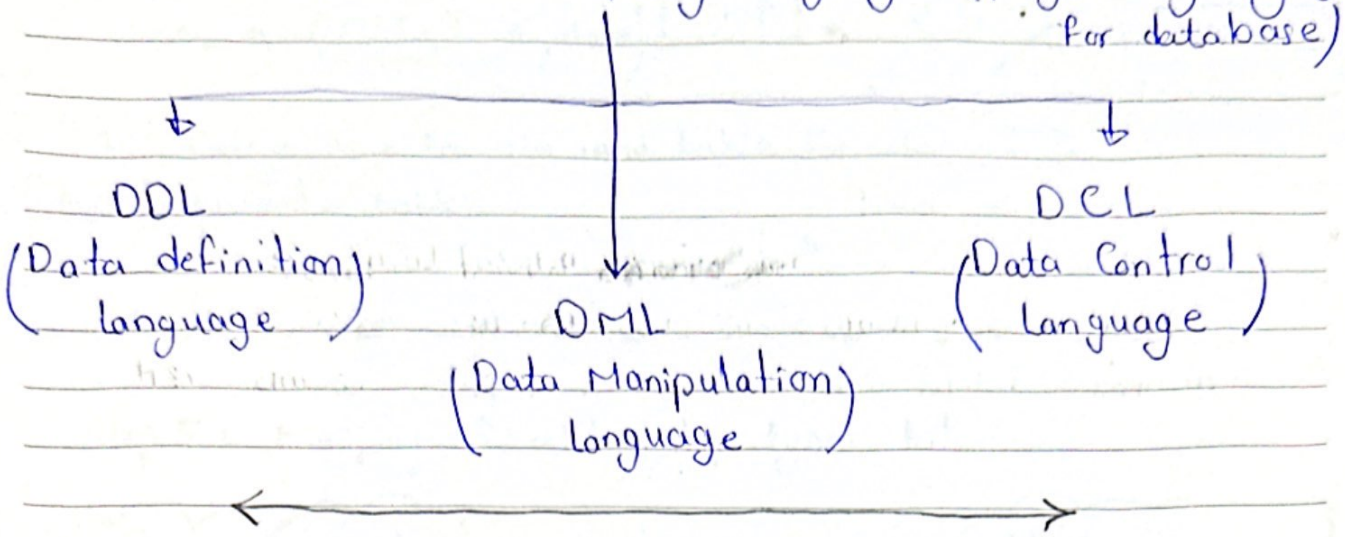
LUCKY

step 6 :- Mapping of ternary relationship (Ternary)

Create new table skilled with (SSN, PNO, Skill_ID)

Chapter 4:- * Physical design *

SQL :- Structured Query Language (programming language for database)



* Data base schema :- • is a group of related objects in a database

• There's one owner of a schema who has access to manipulate the structure of any object in the schema

• A schema does not represent a person, although the schema is associated with a user that resides

* Data Types :- • it determines the type of data that can be stored in a database column.

• The most commonly used data types are :-

1. Alphanumeric
2. Numeric
3. Float data type
4. Data and Time
5. Characters
6. Variable character
7. Integers

LUCKY

• Database constraints:-

↳ Restrictions on Database table or object to help maintain integrity of data

Examples:-

[1] Primary key:- → Not Null, Unique

[2] Not Null:- This enforces a field to always contain a value, which means that you can not insert a new record or update a record without adding a value to this field.

[3] Unique key:-

↳ ensures that all values in the column are different

↳ Both Unique key and Primary key constraints provide guarantee for uniqueness for a column or set of columns.

[4] Referential Integrity (Fk):-

work on → insert:- add in Parent then child

↳ Delete :- delete Fk (child) then Pk (Parent)

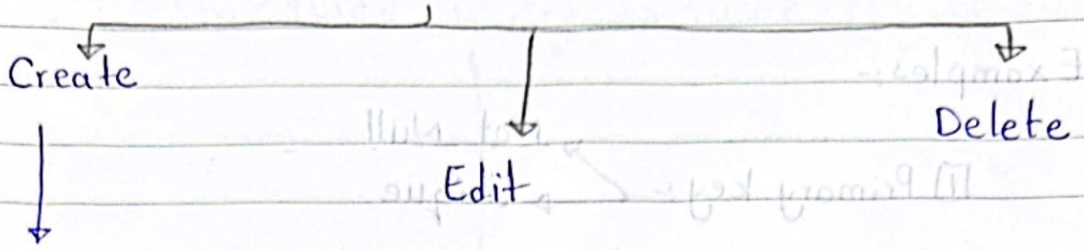
[5] check :-

↳ customize Salary for employee salary is between 1000 and 12000

↳ will check values of the column are within range

LUCKY

- Data Definition Language (DDL)
  - Number of commands are responsible for the structure of Database objects



- CREATE TABLE Students (ID Number PRIMARY KEY, ...etc)

Add column:-

- alter table students add postal_code number

Remove column:-

- alter table students drop column country

Remove table students

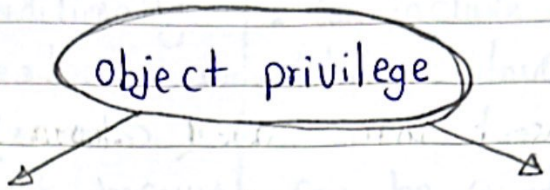
- drop table students

Data Control language (DCL)

Commands:- 1. GRANT command
2. REVOKE command

* use a command that gives privilege access to data

- System privilege
- object privilege



Create privilege (GRANT)	Remove privilege (REVOKE)
• GRANT SELECT ON TABLE employee TO Ahmed,	• REVOKE ON TABLE department FROM Mary;
• GRANT ALL ON TABLE department TO Mary, Ahmed;	• REVOKE ALL ON TABLE department FROM Mary, Ahmed;
• GRANT SELECT ON TABLE employee TO Ahmed WITH GRANT OPTION;	

✱ chapter 5 ✱

Data Manipulation Language (DML)

[1] INSERT Command :-

i- Insert into employee (Fname, Lname, SSN, Address, ...etc)
Values ('Ahmed', 'Hassan', 102672, '8/8/1988', ...etc)

- if values are names or date put within ' '
- insert into table (columns) then values

ii- Insert into employee Values ('Ahmed', 'Hassan', ...etc)

- insert values directly

iii- Insert into employee (Fname, Lname, SSN)
Values ('Israa', 'Abdelghany', 102661)

- insert specific columns only

[2] Update Command :-

Condition

→ update employee set salary = 1200 where SSN = 10622

→ update employee set salary = 1200, dno = 10 where SSN = 10264

3] Delete Command:-

→ delete from employee where ssn = 102672

TRUNCATE	Vs	DELETE
• Deletes all Data but keeps the structure of the table.		• Deletes all data but keeps the structure of the table (if it doesn't have WHERE clause).
• Deletes data unconditionally (doesn't have where clause).		• Can include where clause to delete data conditionally
• Can't be rolled back because it's a DDL statement.		• Can be rolled back (undo) science it is a DML statement.
• De-allocates the physical memory assigned to data.		• keeps the physical memory assigned to data until a commit or roll back is done

4] SELECT Command:-

→ Select Dname, Dnum, [MGRstart date] (columns)
from departments (table)

• if the column name has space in it put in [ ]

→ select *
from departments
where mgrssn = 223344 → display specific row

LUCKY

Comparison & Logical operators:-

* select fname
  from employee
  where salary >= 1500 and salary <= 2500 (OR) where salary between 1500 and 2500.

OR

* select SSN, Fname
  from employee
  where superssn = 321654 or superssn = 223344 (OR) where superssn IN(321654, 223344)

Single row operator — multy row operator

Like operator:-

. = is used for exact value but like operator match pattern

ex:- select *
from employee
where fname like '?o*'

? replace one char and * replace zero or more char

Select *
from employee
where fname like 'Ahm?d' Ahmed, Ahmad

Alias operator (as)

* select fname, salary * 0.1 as Bonus
  from employee

* select don't affect the dat it only displays values since there's no column called saldry * 0.1 so we used as (alis) it will display 0.1 of salary but not added to the table

* select fname + ' ' + lname as [Full Name]
  from employee
  where salary * 12 > 10 000

* will display Nomes of employees whose annual salary > 10 000 (First + last name in column called Full Name)

Order by:-

* select Fname, SSN
  from employee
  order by fname

* defult sorting is ascendingly

* select *
  from employee
  order by dno asc, Salary desc

Distinct ← تمنع التكرار في البيانات

```
*select distinct Dno
 from employee
```

Inner join:- find data from multible columns with key (رابط)

```
*select fname, dname
 from employee e, departments as d
 where e.dno=d.dno
```

*called = join

```
select fname, dname
from employee e inner join departments as d
on e.dno = d.dno
```

(on) Join condition

Outer, full join ⇒ view all data even if not fully matched in columns (if data missed or empty in any)

```
select fname, dname
from employee e left outer join departments d
                  right
                  full
on e.dno = d.dno
```

left shows ← left hand side (employee)
right " → right " " (departments)
Full " both

example

	DP4
Amr	
Ahmed	DP2

missed will not be shown in the results of inner join

LUCKY

Self join shows data from same table and column

```
select e.fname, s.fname
from employee e, employee s
where e.superssn = ssn
```

---

minimum, maximum and count functions :-

```
select count(ssn) as emp, count(salary) as sal
from employee
```

*returns number of elements in the column (count)

```
select max(salary) as max, min(salary) as min
from employee
```

*return min and max in column

---

sub_queries:- (nested query)

```
select *
from employee
where salary >(select salary from employee
               where fname = 'Ahmed' and lname='Ali'
```

displays the employees whose salary is greater than Ahmed Ali

---

Group_by & Having :- divide the table into chunks having given requirements

```
select avg(salary)
from employee
group by dno
having max(salary) >1800
```

LUCKY

Chapter 6

Views

- A view is a logic table based on a table or another view
- A view contains no data of its own, but it is like a window through which data from tables can be viewed or changed
- The tables on which a view is based are called base table
- The view is stored as a select statement in data dictionary.

Ex:-

```
CREATE VIEW vw_work_hrs
AS
select Fname, Lname, Pname, Hours
From employee, project, work on
where SSN = ESSN and PNO = PNUMBER
```

Check option :- checks where condition is valid

EX:-

```
CREATE VIEW Suppliers
As
select *
from suppliers
where status >15
with CHECK OPTION;
```

Modifying a Views:-

- Create or replace —Syntax→ CREATE OR REPLACE VIEW view_name
- Remove —Syntax→ DROP VIEW views-name

Advantages of views:-

1. Restrict data Access
2. Make Complex queries easy
3. Provide data independance
4. Present different views of the same data

Types of views:-

Feature	Simple views	Complex views
Number of tables	one	one or more
Contain functions	No	yes
Contain groups of data	NO	Yes
DML operations through a view	Yes	Not always

Indexes:-

- Use indexes to solve the following Problems:- ① data is not sorted
  ② data is scattered

- why indexes?

1. They are used to speed up the retrieval of records in response to certain search conditions
2. May be defined on multiple Columns
3. Can be created by the user or by the DBMS
4. Are used and maintained by the DBMS

- But causes overhead on DML (Insert, Update, delete) disadvanta

Index creation Guiddines

⇒ Create index when:-
- Retrieving data heavily from table
- Columns are used in search conditions and joins
- Columns contain large number of nulls.

⇒ Do not create an index when:-
- Table is updated frequently

* Create and remove index syntax:-

⇒ CREATE INDEX index_name on table_name (column_name)

⇒ DROP_INDEX index_name

Chapter 7:- Normalization

- Normalization of data is a process that takes a series of tests (Normal forms) to Ⓐcertify the goodness of a design and thus to:-

1. minimize redundancy (duplication of data
2. insert anomalies
3. update "
4. delete "
5. frequent Null values

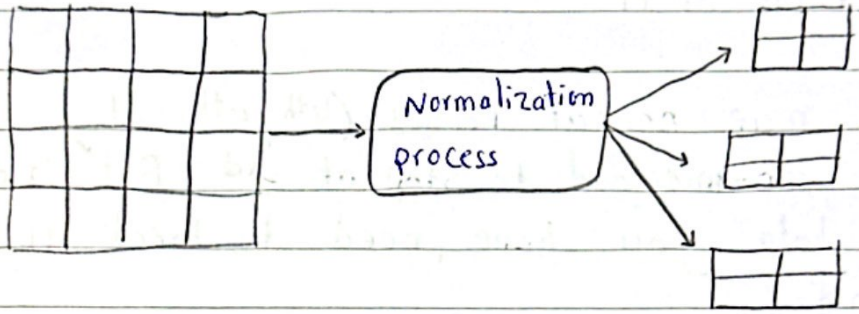
Ⓑ Use normalization to another method for create a database design

Functional dependency:-

A constraint between two attributes (columns) or two set of columns

it has 3 types:-
- fully => fully depend on the primary key
- partially => depend on one of the primary keys (half)
- Transitive => depend on attribute depend on prime key

Normalization can be defined as the process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations.



Normal Form :- Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form.
- first normal form
- Second normal form
- third normal form

First normal form if doesn't have
- Multivalued attribute
- Repeating group
- Composite attribute

* if it has any of them we can break the table to reach the first normal form

LUCKY

Second normal Form → achieve First normal form ✓
→ doesn't have partial dependency ✗

non key attribute depends on key

- Can break the table and create new table of the partial dependent attributes

Third normal form → achieve 2nd normal form
→ doesn't have transitive dependency

- if there's transitive dependency we can create a new table of it

* there are more normal forms (4th, 5th ..etc) but it's recommended to stop at 3rd. But it's dependent on the data you have, need to break the relations more or not

For questions and comments :-

LinkedIn :- www.linkedin.com/in/israa-abdelghany-4872b0222
Gmail :- israaabdelghany9@gmail.com

LUCKY