Lec1:

File systems

- sequential files
- random files

bit → 8 bits byte (char) → set of bytes (field) → set of related fields (record) → set of related records

or attribute (column)

or row

related records: File system: File | database: table | excel: sheet

ID	name	address	gender	grade
1053211	Ali	Ramallah	M	81
1063521	Samir	Jeruselem	M	75
1053211				

stop (repeated key)

⇒ set of related tables "Database system"

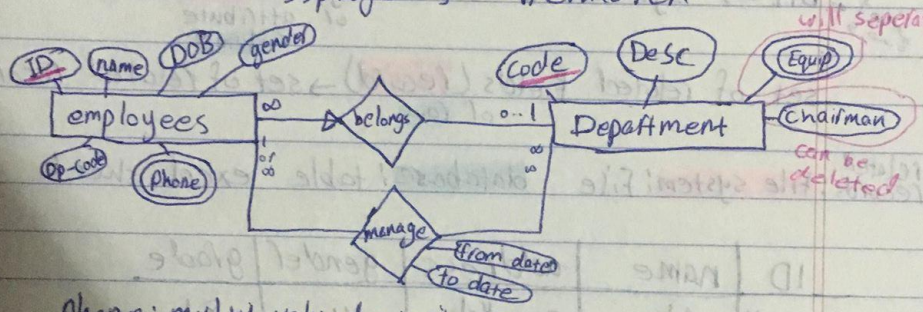
DBMS: DataBase Management System

1. Identify entities. (Ex: company: employees, projects, departments).
2. Find attributes of each entity.
3. Find the relations between the entities.
4. Schema.
5. Create table.

Lec 2:

DBMS: 1. entities 2. attributes 3. relations 4. schema 5. implementation

⇒ comany: 1. employees 2. Department 3. projects 4. children



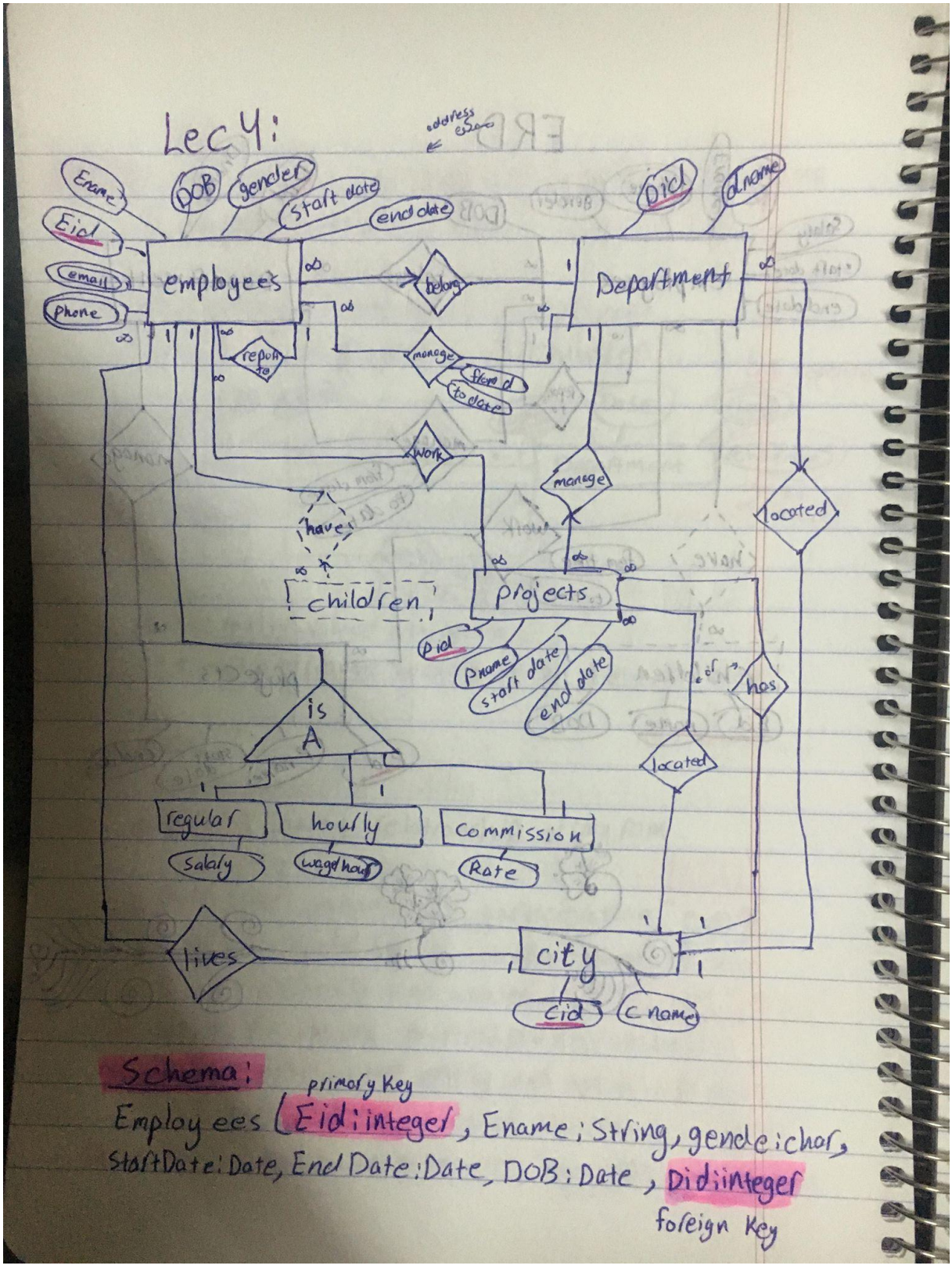
phone: multy-valued attribute

age: not attribute because it can be calculated.

Lec 3:

Lec 3 notes:

- ERD: Entity relationship diagram
- Bold Line: means full participation "every employe should be in a department" if not 0...1 department was written. ← employee with no department (0..1)
- not Bold: means partial participation
- Dashed Line: weak entity and weak relation "full independency on other entities or relations"
- report to: can be replaced with "supervise" أشرف one-to-many.

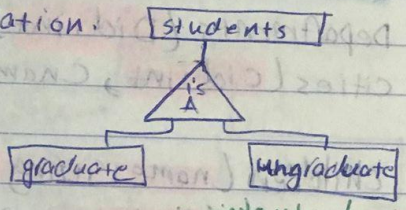


P → primary key
f → foreign key

Department (Did: integer, Dname: string)
projects (Pid: integer, Pname, string, Did: integer, startDate: Date, endDate: Date)

Lec 5:

- non overlap: no intersection between subclasses.
- overlap: intersection between subclasses.
- * only in "is A" relation.



Schema:

Employees (Eid: integer, Ename: string ..... , Did: integer, cid: integer, Eid1: integer) → : new attribute

(only single valued attributes)

(المتزوج)

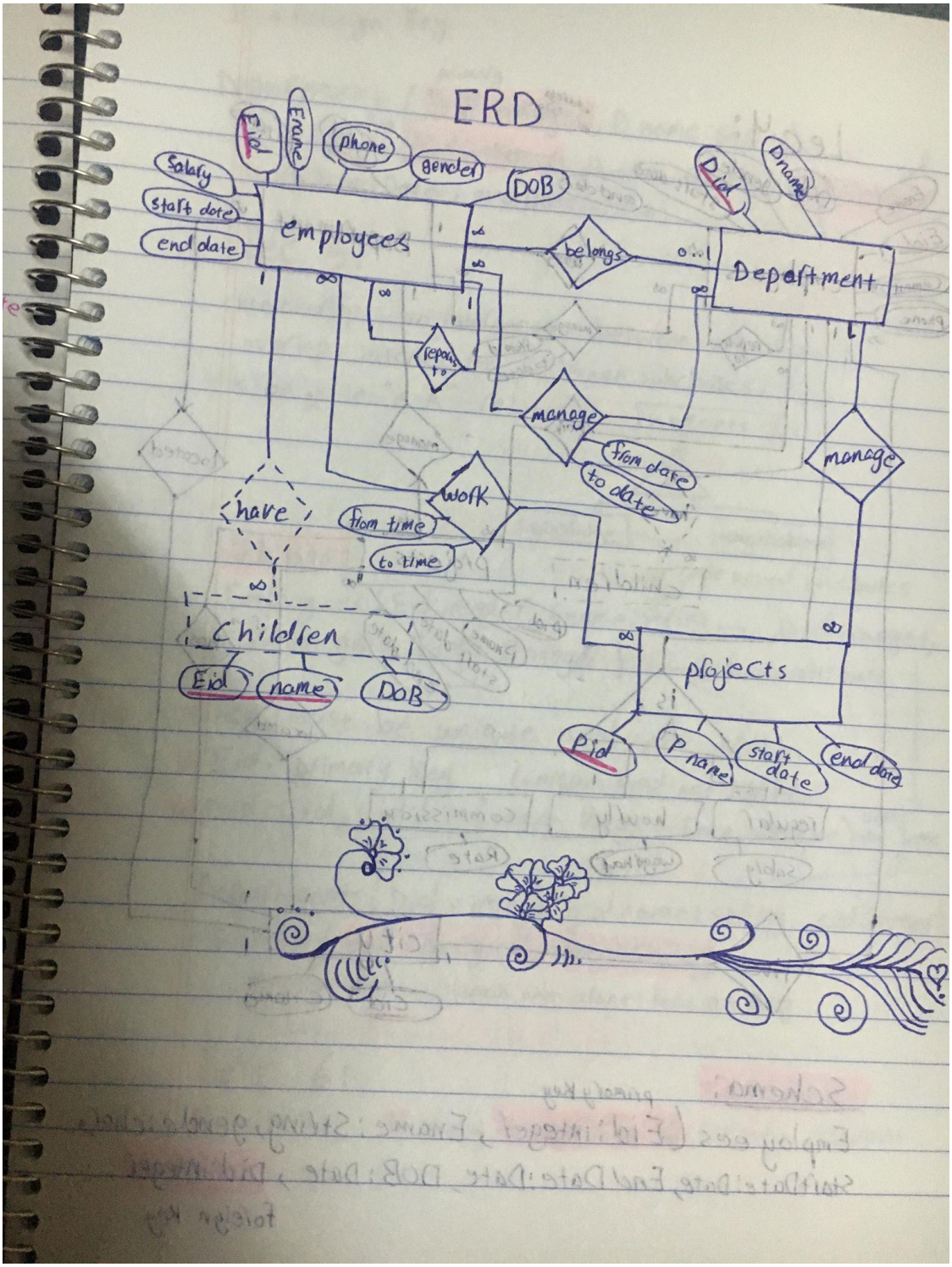
- * key: must be unique and not NULL.
- * Eid: primary key (unique and not NULL)
- * Did, cid, Eid1: foreign keys (not unique) (always the same)

Departments (Did: integer, dname: string, cid: integer)
children (Eid: integer, name: string, ...)

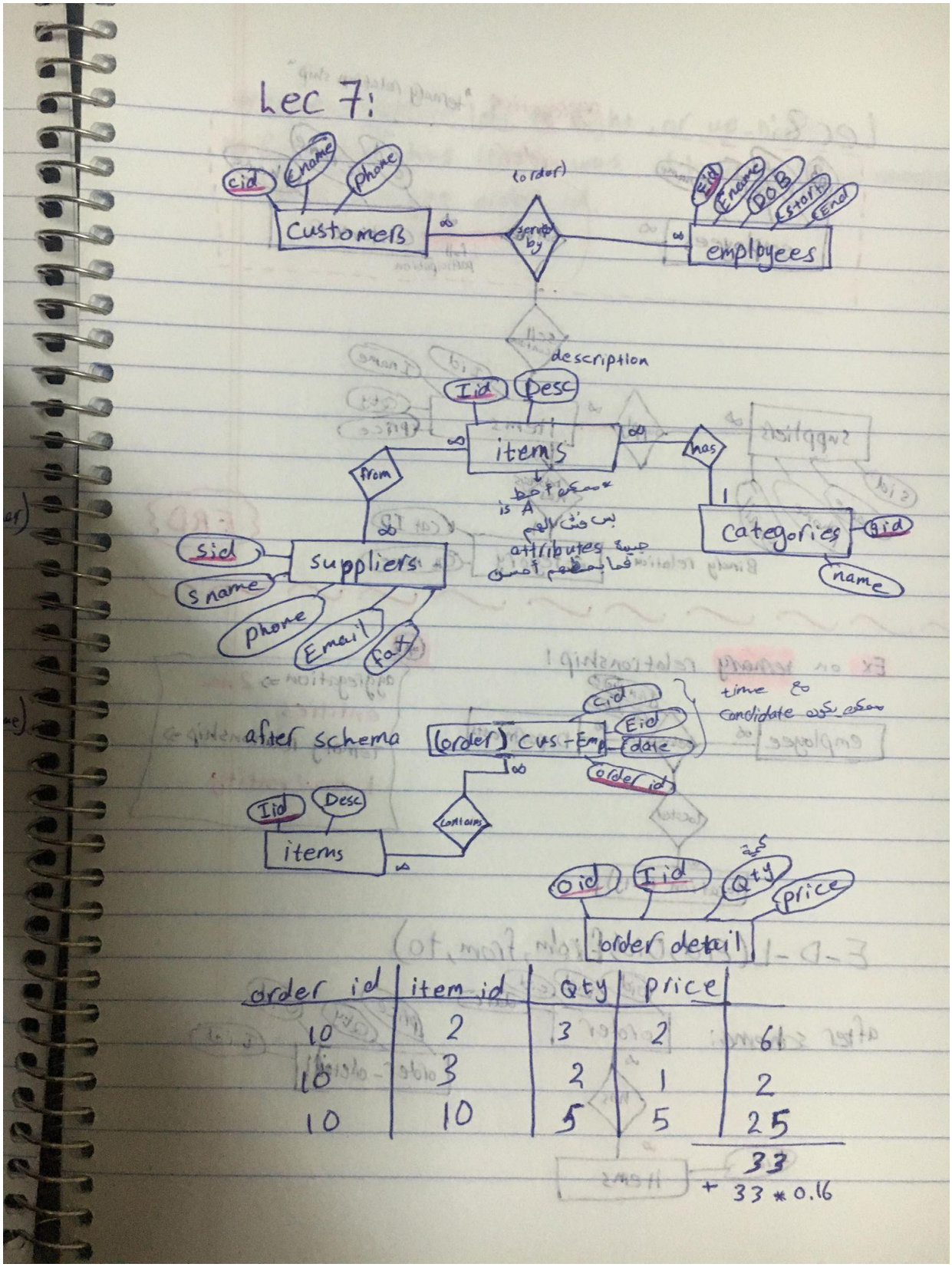
together: key — each one alone: foreign key

Lec 6:

Relations: 1, binary relations 2, ternary relations



Lec 7:



order id	item id	Qty	price	
10	2	3	2	6
10	3	2	1	2
10	10	5	5	25
				33
				+ 33 * 0.16

* many_to_one relations: foreign keys
* many_to_many relations: new entity
* multy-valued attributes: new entity
* one_to_one: entity

P f

Schema:

Employees (Eid:int, Ename:String, ---, Did:int, Eid:int, cid:int)
Departments (Did:int, Dname:String, cid:int)
cities (cid:int, Cname:string)

(weak) children (name:String, Eid:int, DOB:Date, gender:char)
projects (pid:int, Pname:String, Did:int)

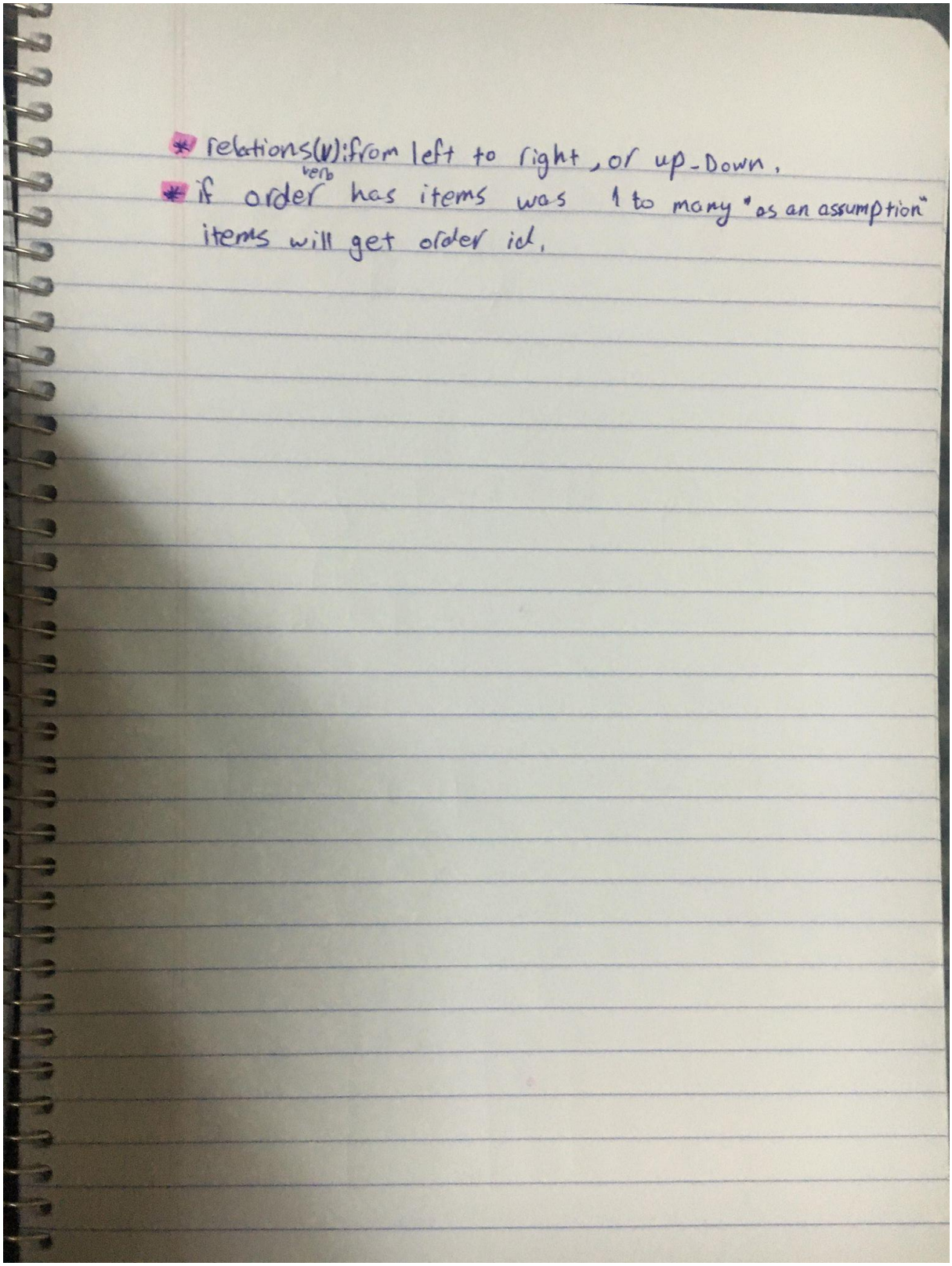
كل واحد لحاله foreign مع بعض Key

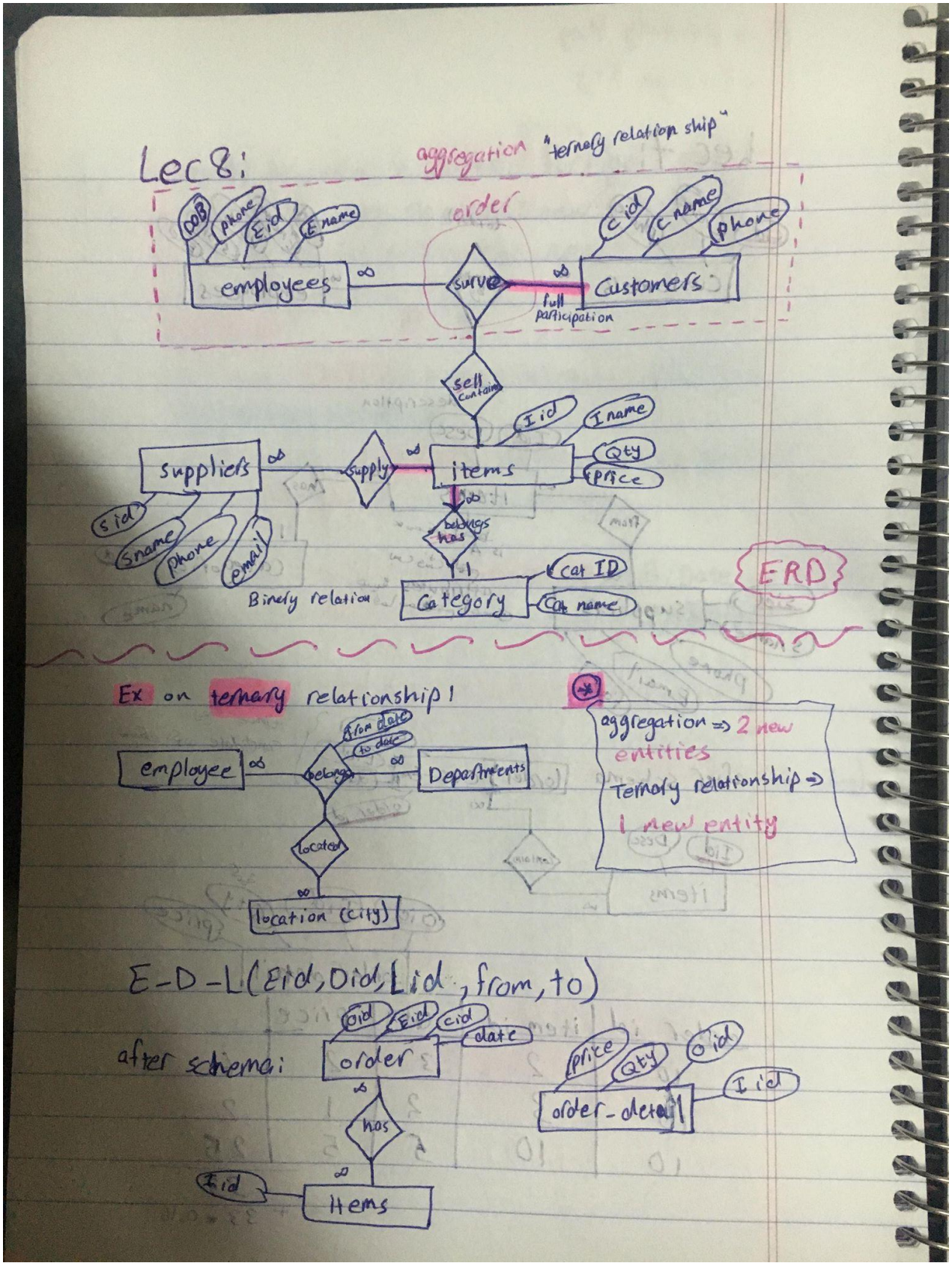
Emp-Dep (Eid:int, Did:int, fromDate, toDate)
* ممكن يكون الموظف بنفس الـ Dep بس تاريخ مختلف
Emp-proj (Eid:int, pid:int, fromTime:Time, toTime:Time)
Time: Date + hour

key entity foreign

regular (Eid:int, salary:float);
hourly (Eid:int, Rate-hour:float);
comission (Eid:int, Rate_com:float);

Emp-phone (Eid:int, Phone:string)
Emp-Email (Eid:int, email:string)





Lec 12:

Chapter 4:

* Relational Algebra "Relational Database"

schema:

Sailors (sid:integer, sname:String, rate:integer, DOB:date)

boats (bid:integer, bname:String, color:String)

Reserves (sid:integer, bid:integer, date:date)

* operators:

1. selection σ Reserves $_{sid=101}$

σ table
Condition

Reserves:

sid	bid	date
110	3	01/01/2018
101	2	15/01/2018
⋮		
101	1	16/03/2018

σ Sailors
DOB = 28/02/2011

2. projection Π

Π sailors
sname, DOB

sid	Sname	rate	DOB
110	Ali	3	01/01/2016
⋮			
150	samir	2	15/01/2015

$\Pi_{sname,rate}(\sigma_{sailors.DOB='28/02/2011'})$

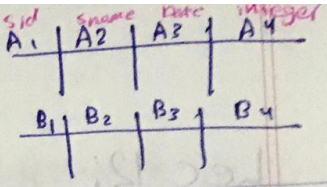
Sname	rate

$\sigma_{DOB='28/02/2011'}(\Pi_{sname,rate})$ ⇐ won't work so $\sigma\Pi$ are not commutitive operators

$\sigma_{DOB='28/02/2018'}(\Pi_{sname,DOB})$ ⇒ will work

in general: $\sigma\Pi \neq \Pi\sigma$

3. set operators:



1, ∪:

2 table compatible must be

1. the same # of attributes
2. the corrosponding attributes have the same domain type.

2, −:

$A - B = \{ x : x \in A , x \notin B \}$

3, ∩

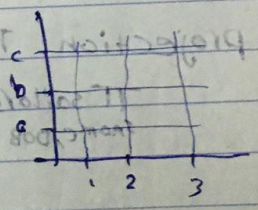
4. ρ(C, sailors) rename

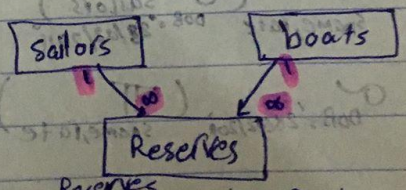
ρ(C(1→ No, 3→ rating, sailors)

5. X "cartesian product"

A = {1, 2, 3} B = {a, b, c, d}

A X B = {(1, a), (1, b), (1, c), (1, d)
(2, a) . . . .
(3, a) . . . }





sid	sname	rate	DOB
101	saleem	2	—
103	Ali	1	—
104	kamil	1	—
105	sajeeb	3	—

Reserves

sid	bid	Date
101	2	—
102	3	—
101	1	—

select *
from sailors s where s.age between 20 and 40

s.age >= 20 and s.age <= 40

select Max(s.age) from sailors s

Ex: Find the sids of sailors with age over 20 who have reserved a red boat.

$$\Pi_{S.sid}\left((Sailors \underset{S.sid = R.sid \text{ and } S.age>20}{\bowtie} reserves) \underset{R.bid = b.bid \text{ and } b.color = red}{\bowtie} boat\right)$$

Ex: Find the names of sailors who have reserved at least 2 boats

at least 1 boat

$$P\left(N\left(\begin{matrix}1\to S_1\\ 2\to S_2\\ 3\to R_1\\ 4\to R_2\end{matrix}\right), \left(\Pi_{sid, sname} Sailors \bowtie \Pi_{sid, bid} Reserves\right)\right)$$

$$P(N1, N), P(N_2, N) \qquad \Pi_{N1.S2}\left(N1 \underset{R_2 <> R_2 \text{ and } N1.S1 = N2.S2}{\bowtie} N2\right)$$

Ex: Find the Sailors who didn't reserve any boat.

$$\Pi_{sname} Sailors - \Pi_{S_2} N$$

SQL

σ_{c} Sailors
c: Sailors.age

select *
from sailors S where S.age>20

σ Sailors
Sailors.age > 20

select S.sname AS "sailor name", S.sid from sailors where S.age>20

sid	sname	rate	DOB	sid	bid	Date
101	Saleem	2		101	2	
101	Saleem	2		102	3	
101	Saleem	2		101		
103	Ali	1		101	2	
103	Ali	1		102	3	
103	Ali	1		101	1	

$$\pi\left(\sigma_{\text{sailor.sid} = \text{Reserves.sid}}(\text{Sailors} \times \text{Reserves})\right)$$

and sid = 101

⋈ ⇒ join x then σ

Lec 13:

Sailors

sid	sname	rating	age
22	Ahmad	7	45
31	Said	8	55.5
58	Ruba	10	35

Reserves

sid	bid	day
22	101	10/10/96
58	103	11/12/73
31	101	11/05/76

cross X

Sailors x Reserves:

#of rows = #of rows in sailors * #of rows in Reserves

= 3 * 3 = 9

Ex: Find the names of sailors who have reserved a red or green boat.

$$\pi_{s.name}\left((sailors \bowtie_{s.sid = R.sid} Reserves) \bowtie_{R.bid = b.bid \text{ and } b.color = \text{"red" or } b.color = \text{"green"}} boat \right)$$

or

بصير اسمه N

$$\rho\left(N, \sigma_{b.color = \text{"red" or } b.color = \text{"green"}} boat\right)$$

$$\pi_{s.name}\left((sailors \bowtie_{s.sid = R.sid} reserves) \bowtie_{R.bid = N.bid} N \right)$$

أفضل

* إذا اختلف اسم sid في R مثلاً sid1 مابزبط الـ natural
لازم أحط S.sid = R.sid1

Ex: Find the names of sailors who have reserved a red and green boat.

$$\rho\left(N, \pi_{s.name}(sailors \bowtie_{s.sid = R.sid} reserves) \bowtie_{r.bid = b.bid \text{ and } boat.color = red} boats\right)$$

$$\rho\left(M, \pi_{s.name}(sailors \bowtie_{s.sid = R.sid} reserves) \bowtie_{r.bid = b.bid \text{ and } b.color = green} boat\right)$$

$$N \cap M$$

Sailors ⋈ Reserves ⇒ Π_{sname}
Sailors.sid = Reserves.sid
Reserves.bid = 103

Ex: Find the names of sailors who have Reserved red boat.

$$\left(S \underset{S.sid=R.sid}{\bowtie} R\right) \underset{R.bid = b.bid \text{ and } b.color = \text{"red"}}{\bowtie} b$$

or ⇒

$$\Pi_{N.sname}\left(P(N, (S \underset{S.sid=R.sid}{\bowtie} R)) \underset{N.bid=b.bid \text{ and } b.color='red'}{\bowtie} b\right)$$

Ex: Find the colors of boats Reserved by Ahmad.

$$\Pi_{color}\left(S \underset{S.sid=R.sid \text{ and } sname='Ahmad'}{\bowtie} R \underset{R.bid=b.bid}{\bowtie} b\right)$$

Ex: find the names of sailors who have reserved at least one boat.

$\Pi_{s.name}$ (sailors ⋈ Reserves) ← natural join
s.sid = R.sid

$\Pi_{s.name}$ (sailors $\underset{S.sid=R.sid}{\bowtie}$ Reserves) ← conditional join

Sailors × Reserves

Sid	Sname	rating	age	sid	bid	day
22	Ahmad	7	45	22	101	10/10/96
22	Ahmad	7	45	58	103	11/12/73
22	Ahmad	7	45	31	101	11/05/76
31	said	8	555	22	101	10/10/96
31	said	8	55.5	58	103	11/12/73
31	said	8	555	31	101	11/05/76
58	Ruba	10	35	22	101	10/10/96
58	Ruba	10	35	58	103	11/12/73
58	Ruba	10	35	31	101	11/05/76

$\sigma_{S.sid = R.sid}$ (Sailors × Reserves) ⇒ σ × : ⊠

sid	sname	rating	age	sid	bid	day
22	Ahmad	7	45	22	101	10/10/96
31	said	8	55.5	31	101	11/05/76
58	Ruba	10	35	58	103	11/12/73

$\sigma_{bid = 101}$ Reserves

sid	bid	day
22	101	10/10/96
31	101	11/05/76

join معناها بو

Ex: Find the names of sailors who have Reserved boat 103.

Ex: Find the names of sailors who have age > 20;

```
select S.sname
from sailors S
where S.age > 20
order by Sname
```

by default asc

$\Pi_{sname}(\sigma_{age>20}(sailors))$

Ex: Find the name of sailor who have reserved red boat.

```
( select Distinct S.sname as sailor_name
  from sailors S, Reserves R, boats b
  where S.sid = R.sid
        and R.bid = b.bid
        and b.colors 'red' ) as N
, b.price + b.price * 0.17 as cost
```

Salior name	cost

Ex:

```
select
from sailors S
where S.sname like 'K%';
```

أي حرف بعده K

'%K%' ← أي حرف قبله وبعده K

SQL

select "Distinct" attributes, expression, calling functions (standard or user functions) renaming to columns

all attributes *
from tables
where condition (join condition) and more other conditions
order by attributes Snames Desc, rating
group by attributes
having group condition

SORT ←

Ex:

select * (S.sname as sailors_names) (R ⋈ S)
natural or equal join
from Reserves R, Sailors S
where R.sid = S.sid
order by S.sname Desc

Ex: Find the sailors names who didn't Reserve any boat.

(select S1.sname from sailors S1.
except
select Distinict S2.sname
from Reserves R, sailors S2
where R.sid = S.sid) as N

أو بحط S.sid عشانها بتجيب تكرار وبكمل
بتشيل Distinict و

except = −

select * from sailors s where s.age between 20 and 40

s.age >= 20 and s.age <= 40

select Max(s.age) from sailors s

* SQL "Structured Query Language"

1. DDL → Data Definition language "create, Drop, Alter"
2. DML → Data manipulation language "insert, update, delete, select"
3. DSL → Data security language "Grant, envoke"

Drop table employee. → table يحذف جميع الـ

delete from employee where eid = 101. يحذف أسطر معينة

select attributes
from table where "condition"
order by ename asc desc
group by ---
having ---

Sailors (sid, sname, rating, age)
boats (bid, bname, color, price)
Reserves (sid, bid, day)

select S.sname from sailors, N where S3.sid = N.sid.

Ex: Find the sailors names who have Reserved at most one red boat.

Ex: select count(*) from sailors S = select count(s.sname) from sailors S

Ex: Find the sailors who have Reserved Red boats and green boats. OR Reserved Red boats or green boats. union

(select distinct S1.sid
from sailors S1, Reserves R1, boats b1
where S1.sid = R1.sid
and R1.bid = b1.bid
and b1.color = 'red')

intersect

(select distinct S2.sid
from sailors S2, Reserves R2, boats b2
where S2.sid = R2.sid
and R2.bid = b2.bid
and b2.color = 'green')

sid	sname	rate	DOB	sid	bid	Date
101	Saleem	2		101	2	
101	Saleem	2		102	3	
101	Saleem	2		101		
103	Ali	1		101	2	
103	Ali	1		102	3	
103	Ali	1		101		

$$\Pi\left(\sigma_{(Sailors.sid = Reserves.sid \text{ and } sid=101)}(Sailors \times Reserves)\right)$$

⋈ ⇒ join x then σ

Lec 13:

Sailors

sid	sname	rating	age
22	Ahmad	7	45
31	Said	8	55.5
58	Ruba	10	35

Reserves

sid	bid	day
22	101	10/10/96
58	103	11/12/73
31	101	11/05/76

cross X

Sailors x Reserves:

#of rows = #of rows in sailors * #of rows in Reserves

= 3 * 3 = 9

Sailors × Reserves

Sid	Sname	rating	age	sid	bid	day
22	Ahmad	7	45	22	101	10/10/96
22	Ahmad	7	45	58	103	11/12/73
22	Ahmad	7	45	31	101	11/05/76
31	said	8	555	22	101	10/10/96
31	said	8	55.5	58	103	11/12/73
31	said	8	555	31	101	11/05/76
58	Ruba	10	35	22	101	10/10/96
58	Ruba	10	35	58	103	11/12/73
58	Ruba	10	35	31	101	11/05/76

$\sigma_{S.sid = R.sid}$ (Sailors × Reserves) ⇒ σ × : ⋈

sid	sname	rating	age	sid	bid	day
22	Ahmad	7	45	22	101	10/10/96
31	said	8	55.5	31	101	11/05/76
58	Ruba	10	35	58	103	11/12/73

$\sigma_{bid = 101}$ Reserves

sid	bid	day
22	101	10/10/96
31	101	11/05/76

Ex: Find the names of sailors who have Reserved boat 103.

Sailors ⋈ Reserves ⇒ Π_{Sname}
Sailors.sid = Reserves.sid
Reserves.bid = 103

Ex: Find the names of sailors who have Reserved red boat.

$$\left(S \underset{S.sid=R.sid}{\bowtie} R\right) \underset{R.bid = b.bid \text{ and } b.color = \text{"red"}}{\bowtie} b$$

or ⇒

$$\rho(N, (S \underset{S.sid=R.sid}{\bowtie} R))$$
$$\Pi_{N.Sname}\left(N \underset{N.bid=b.bid \text{ and } b.color='red'}{\bowtie} b\right)$$

Ex: Find the colors of boats Reserved by Ahmad.

$$\Pi_{color}\left(S \underset{S.sid=R.sid \text{ and } Sname='Ahmad'}{\bowtie} R \underset{R.bid=b.bid}{\bowtie} b\right)$$

Ex: find the names of sailors who have reserved at least one boat.

$\Pi_{s.name}$ (sailors ⋈ Reserves) ← natural join

$\Pi_{s.name}$ (sailors ⋈ Reserves, $s.sid = R.sid$) ← conditional join

EX: Find the names of sailors who have reserved a red or green boat.

$$\Pi_{S.name}\left((Sailors \bowtie_{S.sid=R.sid} Reserves) \bowtie_{R.bid=b.bid \text{ and } b.color="red" \text{ or } b.color="green"} boat\right)$$

or

$$\rho(N, \text{a boat}_{b.color="red" \text{ or } b.color="green"})$$

$$\Pi_{S.name}\left((Sailors \bowtie_{S.sid=R.sid} reserves) \bowtie_{R.bid=N.bid} N\right)$$

* إذا اختلف اسم sid في R مثلاً sid2 ما بينفع الـ natural لازم أحط S.sid = R.sid2

EX: Find the names of sailors who have reserved a red and green boat.

$$\rho(N, \Pi_{S.name}(Sailors \bowtie_{S.sid=R.sid} reserves) \bowtie_{r.bid=b.bid \text{ and } boat.color="red"} boats)$$

$$\rho(M, \Pi_{S.name}(Sailors \bowtie_{S.sid=R.sid} reserves) \bowtie_{r.bid=b.bid \text{ and } b.color="green"} boat)$$

$$N \cap M$$

Ex: Find the sids of sailors with age over 20 who have reserved a red boat.

$$\pi_{s.sid}\Big((\text{sailors} \bowtie_{\substack{s.sid = R.sid \\ \text{and} \\ s.age > 20}} \text{reserves}) \bowtie_{\substack{r.bid = b.bid \\ \text{and} \\ b.color = \text{red}}} \text{boat}\Big)$$

Ex: Find the names of sailors who have reserved at least 2 boats

at least 1 boat:

$$\rho\Big(N, \big(_{\substack{1 \to s_1 \\ 2 \to s_2 \\ 3 \to R_1 \\ 4 \to R_2}}, \big(\pi_{sid, sname} \text{sailors} \bowtie \pi_{sid, bid} \text{Reserves}\big)\big)\Big)$$

$$\rho(N1, N), \rho(N2, N) \quad \pi_{N1.S2}\Big(N1 \bowtie_{\substack{R_2 <> R_2 \\ \text{and} \\ N1.S1 = N2.S2}} N2\Big)$$

Ex: Find the sailors who didn't reserve any boat.

$$\pi_{sname} \text{Sailors} - \pi_{S_2} N$$

SQL

$\sigma_{\substack{c \\ \text{sailors.age}}}$ sailors

$\sigma_{\text{sailors.age} > 20}$ sailors

```
select *
from sailors S where S.age>20

select S.sname AS "sailor name",
S.sid from sailors where S.age>20
```

select * from sailors s where s.age between 20 and 40

s.age >= 20 and s.age <= 40

select Max(s.age) from sailors s

* **SQL** "structured Query Language"

1. DDL → Data "Definition language" create, Drop, Alter"
2. DML → Data manipulation language" insert, update, delete
3. DSL → Data security language "Grant, envoke"

select :

Drop table employee. → table نحذف جميع ال

delete from employee where eid = 101. نحذف أسطر معينة

select attributes
from table where "condition"
order by ename asc des
group by ---
having ---

sailors (sid, sname, rating, age)
boats (bid, bname, color, price)
Reserves (sid, bid, day)

Ex: Find the names of sailors who have age > 20;

```
select S.sname
from sailors S
where S.age > 20
order by sname
```

by default asc

$$\pi_{sname}(\sigma_{age>20}(sailors))$$

Ex: Find the name of sailor who have reserved red boat.

```
(select Distinct S.sname as sailor_name
 from sailors S, Reserves R, boats b
 where S.sid = R.sid
       and R.bid = b.bid
       and b.color = 'red' ) as N
, b.price + b.price * 0.17 as cost
```

Salior name	cost

Ex:

```
Select
from sailors S
where S.sname like 'K%'
```

أول حرف K

أي حرف بأسم K ← '%K%'

SQL

select "Distinct" attributes, expression, calling functions (standard or user functions) renaming to columns

all attributes *
from tables
where condition (join condition)
and more conditions
SORT ← order by other attributes: Snames Desc, rating
group by attributes
having group condition

EX:

select * (S.sname as sailors_names) (R ⋈ S)
natural or equal join
from Reserves R, Sailors S
where R.sid = S.sid
order by S.sname Desc

EX: Find the sailors names who didn't Reserve any boat.

(select S1.sname from sailors S1.
except
select Distinict S2.sname
from Reserves R, sailors S2
where R.sid = S.sid) as N

أو بحط S.sid عشانها بعبر تكرار وبكمل
بنفس حالها Distinict و

except ≡ −

select S.sname from sailors, N where S3.sid = N.sid.

Ex: Find the sailors names who have Reserved at most one red boat.

EX: select count(*) from sailors S = select count(s.sname) from sailors S

EX: Find the sailors who have Reserved Red boats and green boats. OR

(select distinct S1.sid from sailors S1, Reserves R1, boats b1 where S1.sid = R1.sid and R1.bid = b1.bid and b1.color = 'red')

union of green

intersect

(select distinct S2.sid from sailors S2, Reserves R2, boats b2 where S2.sid = R2.sid and R2.bid = b2.bid and b2.color = 'green')

Nested Query:

Ex: Find the names of sailors who have reserved red boat.

```
select  S.sname
from sailors S, Reserves R, boats B
where  S.sid = R.sid
          and
       R.bid = B.bid
          and
       B.color = 'red'
```

better solution:

```
select S.sname
from sailors S, Reserves R
where S.sid = R.sid
          and
R.bid in (select B.bid
          from boats B
          where B.color = 'red')
```

another solution:

```
select S.sname
from sailors S
where s.sid in (select R.sid
                from Reserves R
                where R.bid in (select B.bid
                                from boats B
                                where B.color = 'red'))
```

Ex: find the names of boats not Reserved at all,

```
Select B.bname
from boats B
where B.bid not in (select R.bid
                    from Reserves R)
```

Ex: Find the names of sailors who have Reserved all boats,

nested correlated query

```
Select S.sname
from Sailor S
where not exist (select B.bid from
                 boats B)
      except (select R.bid
              form Reserves R
              where R.sid=S.sid)
```

```
select
from sailors S, Reserves R
where S.sid=R.sid                    "join conditions"
      or, and, <, <=, >, >=, <>, union, Intersect, except
      S.rating between 10 and 20
      S.rating >=10 and S.rating <=20
```

{ like
in, not in
exist, not exist }

Ex: Find the names of sailors who have reserved boat 105.

```
Select S.sname
from sailors S
where S.sid in (select R.sid
                from Reserves R
                where R.bid = 105)
```

Ex: Find the names of sailors who have reserved all boats.

```
select S.sname
from sailors S
where not exist (select b.bid
                 from boats b)
                 except
                 (select distinct R.bid
                  from Reserves R
                  where R.sid = S.sid)
```

*not exist True if the output of next query empty set. False if the output

of the next Query, not empty set.

* exist is True if the next Query is not empty, False if the next Query is empty.

Exi: select S.sname
from sailors S,
where exist (select distinct R.bid
from Reserves R,
where R.sid = S.sid)

⇒ to choose the names of sailors who reserved at least one boat.

* if we replaced exist with not exist: to choose the name of sailors who didn't reserve any boat.

Ex: select S.sname
from sailors S
where S.sname like 'B%'; اسمهم أول حرف B
like 'B_%'; العرف الأول B و أقل اشي الاسم حرفين
like '%B'; آخر حرف B و على الأقل حرفين
like 'haytham%'
→ like '%tham%' إذا مش متأكد من التهجئة

"- ممنوع يكون فاضي"

Ex: Any ⇒ ∃ some
All ⇒ ∀ all

* select S.sname
from sailors S
where S.age >= Any (select S2.age
from sailors S2)

⇒ إذا المجال = بطلعهم جميعهم
⇒ ما بطلع أصغر واحد

* select S.sname
from sailors S
where S.age >= Any (select S2.age
from sailors S2)

⇒ أكبر شخص
⇒ "empty" ما بطلع اشي = إذا بدو الـ

* select S.sname
from sailors S
where S.age <= Any (select S2.age
from sailors S2)

⇒ ما بطلع أكبر واحد

* Aggregate functions (built in functions)

1. count
2. Avg
3. sum
4. min
5. max

```
select count (s.sid)
  from sailors s
```

sid	sname	rating	age
–	–	–	–
–	–	–	–
–	–	–	–

```
select count (distinict R.bid)
from Reserves R
```

sid	bid	Day
101	10	–
102	11	–
105	10	–
101	7	–

Ex: How many times sailor 105 Reserves boats

```
select count (*) as times_105   (or r.bid)
from Reserves R
where R.sid = 105
```

or:

```
select s.sname
from sailors s
where (select count (*)
       from Reserves R
       where R.sid = s.sid)
                     105
```

Ex:

```
select S.sname
from sailors S
where S.sid not in (
        select distinct R.sid
        from Reserves R)
```