TOGETHER WE CAN ACHIEVE MORE

COURSE NAME: Introduction to Database

Topic: Final Term (Spring 2024-25)

NOTES BY

Nishat Tasnim Ema & Anindo Mahmood

Final-Term

Introduction to Database

Creating Views

outline: Describe, create, retrieve, Alter, insert, update, delete and drop.

View (objects): Logically represent subsets of data from one or more tables.

Views used to:

restrict database access

make complex query easy

allow data independence

present different views of the same data

Simple view:

Number of table → one

contain function → no

contain groups of data → no

DML through view → yes

```
select * from salgrade
select grade from salgrade where hisal>1200
create view gradevu1200 as select grade from salgrade
where hisal>1200
system: grant create any view to username
```

```
describe gradevu1200
select * from gradevu1200
drop view gradevu1200
```

Complex view:

Number of table → one or more
contain functions → yes
contain groups of data → yes
DML through view → Not always

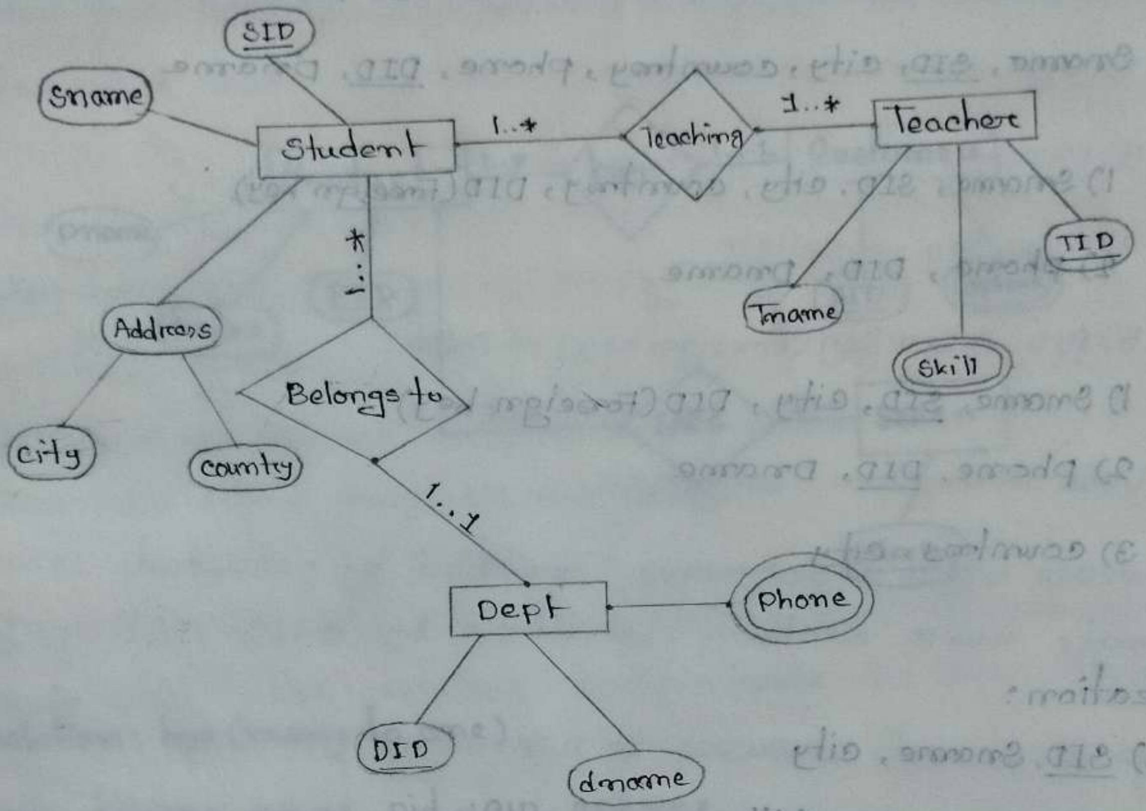
```
select e.ename, e.deptno, d.loc, d.dname
from empe, dept d
where e.deptno(+) = d.deptno
```

```
create view name_and_location as select e.ename,
e.deptno, d.loc, d.dname from emp e, dept d
where e.deptno(+) = d.deptno
```

```
describe name_and_location
select * from name_and_location
drop view name_and_location
```

```
create view sid_sname_srid as select s.sid, ssname,
d.srid from student s, Studentd
where s.sid = d.srid
```

Normalization:



Relation (Teaching) -> many to many

UNF: SID, Sname, address, city, country, TID, Tname, skill

1NF: SID, sname, city, country, TID, Tname, skill

2NF: 1) SID, sname, city, country

2) TID, Tname, skill

3) TID, SID (Foreign key)

3NF: 1) SID, Sname, city

2) TID, Tname, skill

3) TID, SID (Foreign key)

4) country, city

Relation (Belongs to) → many to one

UNF: Sname, SID, Address, city, country, phone, DID, Dname

1NF: Sname, SID, city, country, phone, DID, Dname

2NF: 1) Sname, SID, city, country, DID (Foreign key)

2) phone, DID, Dname

3NF: 1) Sname, SID, city, DID (Foreign key)

2) phone, DID, Dname

3) country, city

Finalization:

1) SID, Sname, city

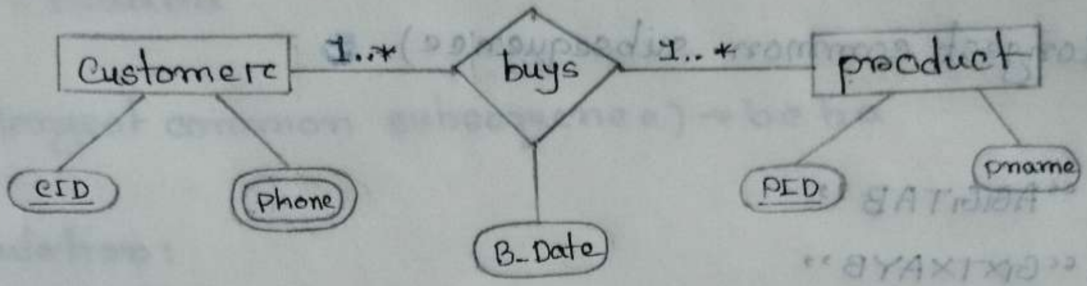
2) TID, Tname, skill

3) TID, SID (Foreign key)

4) Country, city

5) Sname, SID, City, DID (Foreign key)

6) Phone, DID, Dname



UNF: CID , Phone , PID , pname , B-Date

1NF: CID, Phone, PID, pname , B-Date

2NF: 1) CID, Phone

2) PID , pname

3) CID, PID(Foreign key) , B-Date

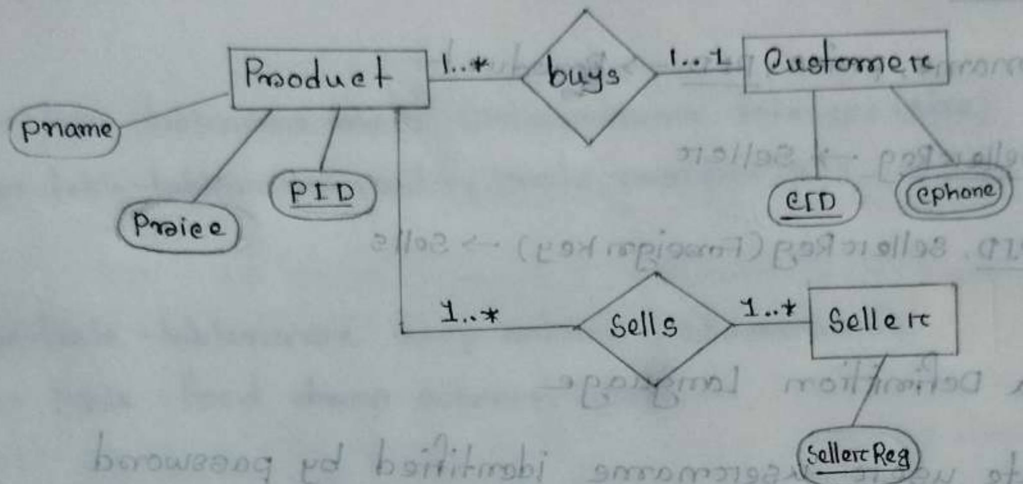
3NF: Same as 2NF

Finalization:

1) CID, Phone

2) PID, pname

3) CID, PID (Foreign key) , B-Date



relation: bys (many to one)

UNF: Pname, price, pid, CID, cphone

1NF: Pname, price, PID, CID, cphone

2NF: 1) pname, price, pid, CID (Foreign key)

2) CID, cphone

3NF: Same as 2NF

relation: sells (many to many)

UNF: Pname, price, PID, SellerReg

1NF: Pname, price, PID, SellerReg

2NF: 1) pname, price, PID

2) SellerReg

3) PID, SellerReg (Foreign Key)

3NF: Same as 2NF

Finalization + Table name

1) pname, price, PID, CID (Foreign key) → Buys

2) CID, cphone → customer

3) pname, price, PID → Product

4) SellerReg → Seller

5) PID, SellerReg (Foreign key) → Sells

DDL: Data Definition Language

Create user username identified by password
Create user canteen identified by c123

grant connect, resource to username
grant connect, resource to canteen

table create:

create table tablename (attributes)
create table food (fID number(3) primary key, fname Varchar2 (5), fprice number(5))

create table using foreign key:
Create table staff (SID number(3) primary key, ssal number(5), fid number(3), constraint fid foreign key (fID) references food (fid))

drop table tablename
drop user username

Alter, rename, drop, insert .delete, update

```
alter table tablename add (column name datatype (SIZE))
alter table food add (rate number(2))
```

```
alter table tablename modify (columnname datatype (size))
alter table tablename modify (rate number (3))
```

```
alter table tablename drop column columnname
alter table food drop column rate
```

```
alter table tablename rename column rate to percentage
alter table food rename column rate to percentage
```

```
rename tablename to newtablename
rename food to fooddetails
```

```
describe fooddetails
describe staff
```

DML: Data manupulation Language

Syntex 2

```
Insert into tablename values ()
Insert into food values (111, 'PIZZA', 230, 7)
```

Syntex 1 (null value)

```
Insert into food (fID, fprice, rate) values (115, 420, 9)
```

Delete rows:

delete from tablename where (condition)

delete from food where fID = 115

Update fields:

update tablename set condition

update food set fname = 'BURGER' where fID = 111

update null values:

update food set fname = ' ' where fID = 111

Table Food:

FID	FNAME	FPRICE	RATE
111	PIZ	230	7
112	BUR	250	8
113	CHW	270	9
114	COC	430	6
115	—	420	—

Table staff:

SID	SSAL	FID
123	13000	111
124	22222	112
125	—	113
126	23111	114
127	32122	115
128	32453	115

Subquery (Single row)

```
select fID, fname, rate from food
where fprice > (select fprice from food where fname = 'PIZ')
```

```
select fID, fname, fprice from food
where rate > (select rate from food where fprice = 480)
```

```
select sid, ssal, fid from staff
where sid < (select SID from staff where fID = 112)
```

```
select SID, ssal, fid from staff
where fid = (select fid from staff where ssal = 28111)
```

Subquery (multiple row)

operator :

In → Equal to any member in list

```
Select ename from emp
where sal in (select min(sal) from emp group by deptno)
```

Any → compare value to each value return by the subquery

```
select ename, empno from emp
where sal > any (select min(sal) from emp
group by deptno)
```

Show the name and hiredate of the employee who was hired before either Jones, blake or smith.

```
select ename, hiredate from emp
where hiredate<any (select hiredate from emp
where ename in ('Jones','Blake', 'smith'))
```

All → Compare value to every value returned by the subquery

```
select empno, ename from emp
where sal > all (select min(sal) from emp group by deptno)
```

Show the name and hiredate of the employee who was hired after Jones, Blake and smith

```
select ename, hiredate from emp
where hiredate > all (select hiredate from emp
where ename in ('Jones', 'Blake', 'Smith'))
```

Joining (using Food and staff table)

Equijoin:

```
select f.fID, s.SID, s.fID, f.fname from food f, staff s
```

Avoid cartesian product: A valid join connection needed

```
select f.fID, s.SID, f.fname, s.fID    from food f, staff s
where f.fID = s.fID
```

Non equi join (using emp and salgrade table)

Show the ename, empno, sal and grade for the employee table

condition: sal in the emp table is in between losal and hisal of salgrade table

```
select e.ename, e.empno, e.sal, s.grade from emp e, salgrade
s where e.sal between s.losal and s.hisal
```

Outer join (using emp table and dept table)

```
select e.ename, e.empno, e.deptno, d.deptno, d.loc
from emp e, dept d
where emp.deptno(+)=d.deptno
```

or

```
select e.deptno, d.deptno, e.ename, d.loc
from emp e, dept d
where d.deptno=e.deptno(+)
```

self joining (using food table)

```
select f.fname, f.fid, d.fprice, d.rate from food f, food d
where f.fID = d.fID
```

self joining concat (using emp table)

```
Select w.ename || 'works under'|| m.ename
from emp w, emp m
Where w.mgr = m.empno
```

Note: mgr in the worker table is empno in the manager table
W.mgr = m.empno

Joining + subquery (using food and staff table)

```
select f.fID, f.price, s.fID, s.ssal, f.rate from
food f join staff s on f.fID = s.fID
where f.fID in (select fID from staff where ssal = 23111)
```

```
select f.fID, f.fprice, s.SID, s.sal from
food f join staff s on f.fID = s.fID
where f.price > any (select fprice food where fname='PIZ')
```

```
select f.fID, f.fprice, s.SID, s.ssal
from food f join staff s
on f.fID = s.fID
where s.ssal < (select ssal from staff where fID = 112)
```

AIUB COURSE SOLUTION-ACS

GROUP: https://www.facebook.com/groups/860221227407452

PAGE: https://www.facebook.com/acsaiub

https://www.youtube.com/channel/UCC3KjA8kstFtM-2CxVr-jcg

AIUB Course Solution - ACS
@acsaiub

AIUB Course Solution - ACS

AIUB Course Solution

- Career Support
- Senior's Opinion
- Faculty Information
- To-let for AIUBians
- Blood Donation
- Club Information

- Book PDFs
- Problems & Solutions
- Assignment Solutions
- Experiment Solutions
- Hand notes
- Video Tutorial