

## DATA BASE

SQL (STRUCTURED Query Language)

SHOW command, Create, Drop and a few others

SHOW databases;

Create Database AIUB;

DROP Database AIUB; database Name

### Data type

#### 1) Numeric data type

i) SMALLINT

ii) INT

iii) BIGINT

iv) FLOAT

v) DOUBLE ( $m, D$ ) size

vi) DECIMAL ( $m, D$ )

#### 2) Character data type

i) char (m)

ii) varchar (m)

iii) text (मात्रा वाले डेटा)

char for (m) मात्रा

#### 3) DATA & TIME DATA TYPE

i) YEAR (YYYY)

ii) DATE (YYYY-MM-DD)

iii) TIME (HH:MM:SS)

## CREATE TABLE

Creating a table

Table को बनाने का **CREATE TABLE** statement

प्रयोग करते हुए **QUA** द्वारा बनाया जाता है

### Syntax

**CREATE TABLE** table\_name

(

column\_name1 data\_type(size),

(m) 2 n

column N data-type(size)

);

Roll	Name	Age	GPA
1	John	20	3.5

**CREATE TABLE** Student

(

Roll int(5),

Name varchar(20),

Age int(5),

GPA double(3,2),

PRIMARY KEY (Roll)

);

constraint PK\_Student

PRIMARY KEY,

## Rename Tables

**RENAME**

command इसका उपयोग एक टेबल का नाम बदलने के लिए

### Syntax

**RENAME TABLE**

~~table\_name old\_name TO new\_name~~

**RENAME TABLE STUDENT TO STUDENTS;**

## DATA INSERTION

Roll	Name	Age	GPA
------	------	-----	-----

Data insert करने का Statement  
235

{INSERT INTO}

Statement को SQL तक  
translate करते हैं।

### Syntax

Roll	Name	Age	GPA
100	Nadim	21	4.00
101	X	22	3.95

INSERT INTO table name (column1, column2 - columnN)

VALUES (value1, value2, ... valueN);

### Example

INSERT INTO Student (Roll, Name, Age, CGPA) VALUES  
(100, 'Nadim', 21, 4.00);

Not

INSERT INTO table name VALUES (value1, value2...);

To Insert multiple records for

Insert INTO table name VALUES

(value1, value2, value3),

(value1, value2, value3),

(-----);

- DDL  $\Rightarrow$  CREATE, ALTER, DROP, RENAME, TRUNCATE
  - DML  $\Rightarrow$  INSERT, SELECT, UPDATE, DELETE, MERGE
- Select Statement

Select statement (এই তালিকা) table থেকে প্রতিষ্ঠা

Symbol	X	101

Select column list

from table name;

Roll	Name	Age	Gender

select Roll

from student;

roll

select Roll, Name,

from student;

Roll	Name

\* সংজুড়ি (অ) form 25

(নথি বিন্দু,

select \*

from student;

## DISTINCT

Duplicate value return এজ নি, table গু সকল value  
 কিন্তু অনিবার্য মানে 'Distinct' keyword প্রয়োগ করলে  
 ১ value মাত্র শব্দটি return করে।

	city
1	Dhaka
2	Dhaka
3	Dhaka
4	Dhaka
5	Comilla
6	Dhaka

## LIMIT

limit keyword এর মানে নির্দিষ্ট নিম্ন তারিখ থেকেই  
 \* Query execute row wise.

প্রথম ৩rd row থেকে চাইলে,

select \*

from student

limit 3;

\* ৩rd row থেকেই প্রদর্শণ করে।

selected

from student

LIMIT 2,3;

select DISTINCT city

from student;

output

city
Dhaka
Dhaka
Comilla

## ORDER by Clause

BUILT IN

কোন ওর্ডের প্রক্রিয়া এবং স্ট্রিক্চুন  
বিষয় মেডেল ক্লাউড এবং বেসিন এর  
অর্ডার অনুসারে 'গৃহীত' করা হচ্ছে।  
by clause এর উপর গৃহীত করা হচ্ছে।

১) Ascending এর মধ্যে তারিখ

(i) select \*  
from student

Order by {Name} column name

NAME
Abul
Babul
:
Z.
Erabut

(ii) select name, age, cgpa  
from student

Order by {cgpa} → মাত্রে cgpa এর অনুসারে গৃহীত

(iii) select \*  
from student

Order by {name, cgpa}; → name & cgpa এর অনুসারে  
গৃহীত।

২) Descending এর মধ্যে তারিখ এবং

select \*  
from student

Order by name{DESC};

## Arithmetic Operator

+ - \* / %

Select 1\*4/6-2

from student;

where age

## Comparison operator

= != > >= < <= BETWEEN

## Logical Operator

AND, OR, IN, NOT, LIKE

(iii) where logical

## WHERE clause

Where clause এর জারামে কি কোন নিয়ম আছে / condition  
কে? উপর লিখি করে দিও কীভুর কো রাখা হবে।

SELECT column list

from table name

Where condition;

select name

from student

Where city = 'Dhaka';

Roll	Name	Gender	City
.			Dhaka
.			ctg
.			Dhaka

Student

## BETWEEN

Select \*  
from student

Where ID BETWEEN 100 AND 110;

where ID NOT  
between 100 and  
110;

## logical operator (IN)

অন্তর্গত OR এর পরিষেতে IN operator মানে।

SELECT \*

From student

Where City = 'Dhaka'  
OR  
City = 'Ctg'

OR

City = 'Noakhali'

OR

City = 'New York';

SELECT \*

From student

Where City IN ('Dhaka', 'Ctg',  
'Noakhali', 'New York');

## NOT IN

Where City NOT IN ('Dhaka', 'Ctg', 'Noakhali');

## LIKE LOGICAL OPERATOR

- ① SELECT \*  
from student  
Where Name LIKE '%N%'; [Name का अंत में N हो रहा।  
यहाँ इसी तरह matter भी है]
- ② SELECT \*  
From student  
Where Name LIKE 'N.%'; [Name का शुरूआती तरफ N हो रहा।  
इसी तरह matter भी है]
- ③ SELECT \*  
From student  
Where Name LIKE '%NA%'; [Name का - प्रथम व दोनों  
अंकों का एक लाला matter जै।  
जिसमें NA नहीं होता]
- ④ SELECT \*  
From student  
Where Name LIKE '\_N%'; [2nd character N हो रहा है]  
[→ 3rd .. N .. n]
- ⑤ SELECT \*  
From student  
Where Name LIKE '%N--'; [3rd character N हो रहा है  
3rd character N हो रहा है]
- ⑥ SELECT ENAME from emp  
Where ENAME LIKE '\_N\_'; [3rd character N हो रहा है  
N का एक और एक character है]

Something + null = null

" + zero(0) = something

### As key word

Column या custom तरीके द्वारा एक keyword

जैसे कोई नहीं

select name, roll - | select name AS 'first name',  
from student; | Roll AS ID | from student;

{ select ename, sal, job, sal\*12 AS Annual\_Salary  
from emp; | "Annual salary"

### SQL Constraints

1. NOT NULL (यह लोग बताते हैं कि NOT NULL क्या है)

(value insert करते हैं)

2. UNIQUE

(Does not allow to insert a duplicate value in a column)

3. PRIMARY KEY = NOT NULL + UNIQUE

### Example

Create Table Emp,

( ID int NOT NULL )

AUTO\_INCREMENT,

Name varchar

NOT NULL,

Salary double (10,2),

PRIMARY KEY (ID)

);

:(Cutting me off) when you see

seafloor spreading ( $\rightarrow$  see Fig. 1)

## UPPER & LOWER

4 T = 0.1 mW

# DELETES FROM EASY

What would you like to do?

~~DELETE~~ form table name

## DELETION STATEMENT

Set NAME = NADIM WITH I = 0 FOR 13

Update Emp

Caracter	Definición	ej.
azules	D	C
NUL	E	W
azules	F	I
azules	Naranja	ID

$$f: \mathcal{C} = \mathcal{D} \setminus \text{Ways}$$

~~000521 = 12500~~ SET

UPDATE Emp

## Update Statement

## ④ Concatenation operator

① select CONCAT (ENAME, job) AS "stu information"

From emp;

② Select ENAME || job

From emp;

student

③ SELECT ENAME || ' IS A ' || job || ' AND HIS

SALARY IS ' || SAL AS 'Emp Information'

from emp;

④ SELECT ename || ' IS A ' || job

from emp;

information

## ⑤ Greatest / Least

Select Greatest (10, 20, 60, 50)

from dual;

## ⑥ Log, Power

Select log (2);

Select pow (2,3);

Grade X

## Count

Table বিশ্বের সকল রেকর্ডের সংখ্যা হলো

```
select count(*)  
from emp;
```

## Max / Min

বড় মানটি দেখাব।

```
select max(sal)  
from emp;
```

## Sum

```
select sum(sal)  
from emp;
```

Avg

```
select avg(sal)  
from emp;
```

## Sub Query

একটি ক্ষেত্রে অন্য ক্ষেত্রের মধ্যে একটি ক্ষেত্র হচ্ছে Sub query

যেমন

→ 

```
select avg(sal)  
from emp;
```

select \*

from emp

→ 

```
select *  
from emp  
where sal > 200.59;  
where sal > 200.59;
```

```
where sal > select avg(sal)  
from emp;
```

## ALTER

- 1) Table  $\rightarrow$  সূচনা করা হওয়া পদ্ধতি
- 2) সূচনা করার নাম পরিবর্তন করা পদ্ধতি
- 3) সূচনা করার delete করা পদ্ধতি

Syntax (1) করার পদ্ধতি

ALTER TABLE table\_name

ADD column\_name datatype [size];

ALTER table emp

ADD phone text(20);

(ii) করার নাম পরিবর্তন পদ্ধতি

ALTER table table\_name

CHANGE old\_column\_name new\_column\_name datatype [size];

Example

ALTER table emp

CHANGE PHONE PHONE\_number text(15);

\* ALTER table emp

Rename to empdetails;

### III) ALTER Delete col

ALTER table table name

DROP column column name;

ALTER Table emp

Drop column phone-number;

### Null value

Select comm  
from emp

where comm = null

Select comm  
from emp

where comm is null

comm is  
not null

### Group BY clause

You can divide rows in a table into smaller groups by using the GROUP BY clause

#### Syntax

Select column group function (column)

from tablename

[where condition]

'group by column name

[ORDER BY column];

select job sum(sal)

from emp

Group by job;

## Truncate

{ Truncate table emp; }

Table ରେ କାହାରୁ କେବଳ ତିଲିଟି ରୁହି ଥାଏ, ଯୁଦ୍ଧ କାମରୁ  
ହାଲୀ ଥାଏ, କାନାରିଗୁଡ଼ୋଟେ ନାହିଁ ଏହି Data insert  
କାହାରୁ

## \* Aggregate

Function ରେ ଆଜିର ନାମର କଲମ  
use କରି



Select  $\max(SAL)$  From emp;

✓ Select  $\max(SAL)$ ,  $\arg(SAL)$ ,  $\sum(SAL)$ ,  
~~MAXSAL~~ from emp;

↳ \* 'Group By' ରେ କାହାରୁ ଏହି କରନ୍ତି କାମରୁ କରିବାକୁ

1) select job, count(\*) From emp Group by job;

2) select job,  $\max(SAL)$  From emp Group by job;

3) select deptno, job,  $\max(SAL)$  From emp  
Group by [Deptno, job];

\* Group by column गो नाम रात्रि रुपी

NVL ~~privileges~~ commission ~~for query~~ the expression NVL

select comm + SAL from emp;  
\* अगर comm null तो sal + comm को null show करें।

select NVL(comm, 0) from emp;

अगर comm null तो comm 0 को replace करें।

select sum(NVL(comm, 2)) from emp;

अगर comm null, तो comm 2 को replace करें।

Function

select ename, length(ename) from emp;  
" " , " (ename) " ;

(\*) select max(sal) from emp where max(sal) > 200; output: error

Aggregate function गो जिसके where clauses use max(sal) जैसी

Q) Aggregate function of where & after Having  
use करते हुए Group by के लिए Having use  
करें 25

X select Deptno, MAX(SAL) From emp

Where MAX(SAL) > 2850 Group by DEPTNO ;

✓ Select Deptno, MAX(SAL) From emp

Group by DEPTNO Having MAX(SAL) > 2850 ;

Select Job, SUM(SAL) From emp

Where Job not like 'S%'

HAVING SUM(SAL) > 4500

Group by job Order by SUM(SAL) ;

B

(pre-which)  $\rightarrow$  (17-Nov-81)

Select hiredate from emp;

Select hiredate  $\rightarrow$  To-CHAR(hiredate, 'DD,mm,YY') $\rightarrow$  (17-11,81)  
from emp;

B

DATE & TIME

Select CURDATE(); / Select sysdate();

Select CURTIME();

Select now(); Day, month, year part

Select ADDDATE('2022-10-8', Interval 5 DAY);

Select subdate('12-11-2022');

Select MAKEDATE(2001, 300);

Select DAYNAME('2022-10-26');

Select Monthname('18-11-2022');

• MONTHS BETWEEN ('01-SEP-95', '11-JAN-94')

(8M+1)

→ 19-6774194

• ADD\_MONTHS('11-JAN-94', 6) → '11-Jul-94'

• NEXT\_DAY ('01-SEP-95', Friday) → '08-SEP-95'

"LAST\_DAY ('01-SEP-95')"

### To-Char Function

- 1) select Hiredate from emp; → 17-Nov-81
- 2) select Hiredate, TO\_CHAR(Hiredate, 'DD-MON-YY')  
from emp; [17-Nov-81]
- 3) Select Hiredate, To\_char(Hiredate, 'DD-Month,YY')  
From emp; [17-November-81]
- 4) Select Hiredate, To\_char(Hiredate, 'DD-Month,YYYY')  
From emp; [17-November-1981]
- 5) Select Hiredate, To\_char(Hiredate, 'DD-Month,Year')  
From emp; [17-November-Nineteen Eighty-one]

6) Select Hiredate, To-Char(Hiredate, 'DY-Month-Year')  
From emp; [TUE-NOVEMBER-NINETEEN EIGHTY-ON<sup>2</sup>]

7) Select hiredate, To\_char(hiredate, DD-MONTH-YYYY)  
From emp; [7-TUE-NOVEMBER-1981]

## Levels of Abstraction

Physical level: Data from location & store ~~city~~, exactly from drive & ~~city~~. Location relates info-  
রুলি মাত্র physical level.

Form of Data (କ୍ଷେତ୍ରି ମୋଟାରେ ଲାଗୁ ହେବାରେ), bytes  
ମାତ୍ରା - ପାଇଁ) ଏ ସିନ୍ଗଲ୍ କ୍ଷେତ୍ରି physical level ଓ ମାତ୍ରା,  
Data structure related ହାଲୁ ମାତ୍ରା info physical level  
ମାତ୍ରା, control by administration, (internal level ଓ ଦରି)

Logical level: describe data type, relationship among the entity and attribute. Table store.

view level

user can't see info hide

### DECODE

Select JOB, SAL, DECODE(JOB, 'Analyst', SAL+20)

'CLERK', SAL+50, SAL) From emp;

Job Analyst salary SAL + 20  
Job Clerk          SAL + 50

### \$

Select SAL, TO-CHAR(SAL, '\$99,999') From emp

### NVL2

NVL2 null value return

Select SAL, COMM, NVL2(COMM, COMM/SAL, SAL)  
From emp;

comm null then SAL print

null at then COMM/SAL print

\* NVL2 string value

string value

\* Nested group function & group by inner function  
↳ If any group has null value then whole row will be null

### Null if

Select mgr, comm, nullif (mgr, comm) From emp;

With mgr, comm as value same as org but null

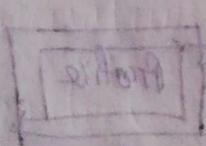
Return mgr, with same as org but null

Parameter return as

filters from maintaining data

Important note for filters

197



Filters shows

interrelated as ii



Fig 200P210

Learning

Fig 200P212

more

more

seas

Fig 200P210

Learning

Fig 200P212

more

seas

Fig 200P210

Achoring

more

more

seas

abide me

PC

Patel

FT

## To SUB query

- Which employees have a salary greater than ~~SMITH~~ SMITH's salary?

select SAL from emp where ename = 'SMITH' ]  
Select ename from emp where SAL > [

#1  
select ename from emp

where SAL > ( select SAL from emp where

where ename = 'SMITH' );

↓ basically ~~in the result~~

select ename, SAL from emp ] assume SAL = 2975  
where SAL > 2975 ;

#2  
select \* from emp (single row sub query)

where SAL = ( select MAX(SAL) from emp ;

#3  
select \* from emp  
where SAL = ( select MAX(SAL) from emp Group by  
Dept\_no ) ;

Output error : single row sub query returns more than one row

\* sub query finds the table to already exist.

select ename, deptno from emp  
where deptno IN (select deptno from dept  
where loc = 'DALLAS');

# Write a query to display the name, deptno and salary  
of employees whose deptno and salary match  
the department number and salary of any  
employee who is earning a commission.

select ename, deptno, sal from emp  
where (deptno, SAL) IN (select deptno, SAL  
from emp where comm is not null);

# Write a query to display the name and salary  
of employees whose salary and commission  
match the salary and commission of any  
employee holding job salesman.

select ename, sal from emp where (SAL, COMM)  
IN (select SAL, COMM from emp where job = 'salesman');

# Write a query to display employees who get higher salary than employee whose employee id is 7782.

Select \* from emp where sal > (Select sal from emp where empno = 7782);

(Select sal from emp where empno = 7782);

# Write a query to display the information of employees whose designation is same as the employee whose employee is 7782.

Select \* from emp where job in (Select job

from emp where empno = 7782);

Multiple column subqueries

Select \* from emp where (sal, Deptno) IN

(Select sal, Deptno from emp where comm is not null);

# Display the second maximum salary

1. Select max(sal) from emp where sal <> (Select max(sal) from emp);

2. Select max(sal) from emp where sal < (select max(sal) from emp);

# Display third maximum salary.

Select max(sal) from emp where  
sal < (select max(sal) from emp where sal <  
(select max(sal) from emp));

# Select all the employees who work in the  
DALLAS.

Select \* from emp where Deptno In

(select Deptno from dept where LOC = 'DALLAS')

# Display all the employees who are earning

more than all managers.

Select \* from emp where SAL > ALL

(select SAL from emp where job = 'Manager');

# Display all the employees who are not located  
at DALLAS.

Select \* from emp where job = 'SALESMAN' and  
Deptno <> (select Deptno from dept where  
LOC = 'DALLAS');

## Single row operator

=, <, >, <=, >=, <>

\* sub query ref single row return two or more result ref comparison by OR in one row  
sub query ref

select ename, sal from emp where sal <> />/  
(select min(sal) from emp);

## Multiple row subquery

IN, NOT IN

\* multiple row subquery giving single row  
operator - in, not in

X select ename, sal from emp where sal =  
(select min(sal) from emp group by deptno);

IN

## Constraints

1) Unique

2) NOT null

3) Primary key (unique + not null)

4) Check

5) Foreign key

6) Default

Table of 6 types of constraint use  
PK, FK, CK, UNI

1. Unique: It prevents value from being unique across duplicate  
value stores. It is used for primary key.

create table student (id int unique,  
name varchar(20),  
age int);

2. NOT NULL:

create table student (  
id int not null,  
name varchar(20),  
age int);

### 3. Default

```
create table student (
```

id int,

name varchar(20) default 'No value inserted',  
age int );

### 4. Check

```
create table student (
```

id int ,

name varchar(20),

age int check (age >= 18)

);

### 5. Primary key

```
create table student
```

```
( id int primary key,
```

name varchar(20)

age int

);

\* Alter table student

Add primary key (id);

\* Alter table student

ADD constraint pk\_student Primary key (id);

Not null → column level

Default → n n

Primary key → Table both

Check → n n

unique → n ..

Foreign key → Table

Alter table student

~~All~~ Drop primary key;

constraint pk\_student

Primary key (id);

## 6. Primary Foreign Key

Create table student (

id int primary key,

name varchar(20);

sec\_id int foreign key Reference section(section\_id)

### ADD constraint

ALTER table student

ADD FOREIGN KEY (sec\_id)

REFERENCE section(sec\_id)

ALTER table student

ADD constraint FK-section Foreign key (sec\_id)

Reference section(sec\_id);

### Drop constraint

ALTER table student

DROP Foreign key FK\_section;

drop foreign key fk\_section

foreign key fk\_section

child constraint fk\_section

# Week → 7, 10 Theory

10

## Normalization

(TUS) Normalization 6.1.5

Redundancy → ~~repetitive data~~ remove repetitive data  
→ ~~data~~ Repetitive data storage → ~~memory~~ memory consumption  
Normalization → ~~use~~ reduce memory consumption & reduce

## Anomalies

1) Insertion

2) Deletion

3) Modification/Updation


## First Normal Form (1NF)


A relation in which intersection of each row and column contains one and only one value.

A relation that contains no multivalued attributes.

\* multivalue attribute composite primary key ~~attribute~~ include

→ ~~total dependency~~  
→ ~~figo~~ ~~figo~~

→ ~~foreign keys~~ ~~different~~ ~~and~~ ~~different~~

→ ~~different~~ ~~and~~

→ ~~one-to-one~~ ~~foreign~~ ~~attribute~~

## 2nd Normal Form (2NF)

mitgliedschaft

A relation will be in 2NF if,

1) A relation is already in 1NF

2) The non-key attributes are fully functionally dependent on the primary key

3) No partial dependency

\* 2NF के नियम किसी economies का नहीं

A	B	C	D
E			

(Fully functionally dependent)

A	C	B
E		

Partial functional dependency

## 3rd Normal Form

A relation will be in 3rd normal form if,

1) It is already in 2nd normal form

2) If there is no transitive dependency in the relation.

~~transitive dependency~~ 3rd normal form

transitive dependency :- यदि एक एक समान value

एक जो Primary key है वह आपके अन्य बाकी को Primary key के तुलना में हो।

## JOINING

यह वा अलगी table को data Cartesian show करता

X Select \* from emp, dept ; → (Cartesian product

in table जो तब तक मिल रही है उसके बाहरी इनमें से एक जो

जो [14x4=56] rows

St.ID	St.Name	cgpa	St.ID	Hobby	Dept
100			100		
101			100		
102			100		
100			102		
101			102		
102			102		
103			102		
100			103		
103			103		

Eff.

1) दोनों टेबलों को मिलाकर एक टेबल का बनाया जाता है।

Equijoin: ~~বেস কোন ক্ষেত্রেই পাস না হবে~~

মান ক্ষেত্রে ~~join~~ table জুড়ে foreign key এর মান  
অন্য table এর primary key join করে দেয়

#1 select \* from emp, dept

where emp.deptno = dept.deptno;

#2 select ename, job, sal, dname, loc from emp, dept

where emp.deptno = dept.deptno;

(match, ename, job, sal emp table এর মান, &

dname, loc dept table এর মান, এক তালে কোন

comment some column এর মান এক তালে কোন মুদ্রণ  
করা ফার

#3

select deptno from emp, dept where  
emp.deptno = dept.deptno

run করা হলো  $\rightarrow$  এক তালে deptno column

emp, dept table জুড়ে column কে run করা

compiler ক্ষেত্রে [column ambiguously defined]

#4

ambiguous problem ~~solved with any table prefix~~

→ use ~~emp~~ 231

(1) Select emp.ename, emp.sal, emp.job, dept.dname,  
dept.loc, dept.deptno from emp, dept where  
emp.deptno = Dept.deptno;

(2)

#5 Select \* from EMP E, DEPT D  
where E.deptno=D.deptno;

(3)

#6 Select E.ename, E.job, D.Dname, D.loc From  
EMP E, DEPT D where E.Deptno = D.Deptno;

Equijoining with AND operator

Select E.ENAM, E.JOB, E.SAL, D.DNAME, D.LOC, D.DEPTN  
From EMP E, DEPT D where  
~~where E.Deptno = D.Deptno AND E.ENAM = 'SCOTT'~~

NON-EQUIJOINING

#1 table GA ~~with~~ primary key ~~as~~ foreign key  
शास्त्र न,

#1  
select ENAME, JOB, SAL, GRADE, LOSAL, HISAL  
From Emp, SALGRADE where  
SAL BETWEEN LOSAL AND HISAL ;

#2  
Select ENAME, JOB, SAL . . .  
From EMP, SALGRADE where  
SAL BETWEEN LOSAL AND HISAL  
AND GRADE = 3;

Problem: select all the departmental information  
for all managers.

Select E.ENAME, E.JOB, D.\* From emp E,  
Dept D Where  
E.deptno = D.deptno AND JOB = 'MANAGER' ;

### Problem:

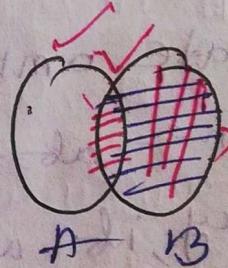
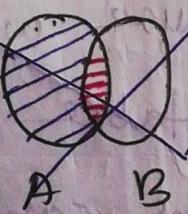
Select all employees who work in DALLAS.

Select ename, job, Dname, loc  
From Emp, Dept

where Emp.deptno = Dept.deptno AND Loc='DALLAS';

### Outer Join

i) Left outer join → It gives the matching rows and the rows which are in left table but not in right table.



⇒ Outer join operator is the plus sign (+)

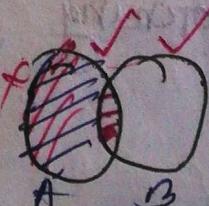
Select e.ename, d.deptno, d.dname From

emp emp e, dept d where

e.deptno = d.deptno (+)

order by e.deptno;

ii) Right outer join :



• self join: can be used to join a table with itself.

Details: In a self join, each ~~at~~ row of the table is joined with itself and all the other rows of the same table.

A self join is mainly used to combine and compare the rows ~~at~~ of the same table.

But whenever we perform self join, it creates ambiguity because we have to use the name of the same table ~~name~~ again. In a query, if we write the same table name twice, it will give an ~~error~~ error. So as to avoid these circumstances, we have to use ~~a~~ 'alias' names for the same table. An alias name simply provides a table with a diff. name for performing such operations.

## VIEW

\* Data hide towards the view user can't see it

Create view empviews50 AS

Select empno, job, ename from emp where deptno = 20;

User can't ename, job, ename show

\* view works like table.

1) select \* from empviews50;

2) select \* from empviews50 where job = 'CLERK'

3) Delete view query

Delete from empviews50 where ename = 'SMITH';

\* view can be drop like table

Drop view empviews50;

Column ALIASES VIEW

Create view empviews50 AS select

Ename as Employee\_name, EMPNO as Employee\_ID,  
SAL as salary, Deptno as Department\_Number from  
emp where Deptno = 30;

- 1) select min(sal) from empview50;
- 2) select min(salary) from empview50;

### To MODIFY view

CREATE OR REPLACE view empview50

(Employee\_name, Employee\_ID, SALARY, Department\_number)  
AS select Ename, Empno, SAL, Deptno from emp  
where Deptno = 20;

To view to user to run DML query (Delete, update)  
(Delete, update, insertion)  $\Rightarrow$  To allow not to delete,  
not's user  $\Rightarrow$  read mode, modify mode  
Create at particular view to  $\Rightarrow$  read only mode  
in the DB.

create or replace view empview50  
(Employee\_name, Employee\_ID, SALARY,  
Department\_number) AS select ename,  
empno, SAL, Deptno from emp where  
Deptno = 20 with read only;

## I) Complex view

মূল view টি joining, aggregate function, group by করার পুরো অংশের সম্মত combination query ফর্ম মূল view create কর এস্ব।

Simple view. সহজে select query ফর্ম মূল view  
create কর এস্ব।

## II complex view

Create view emp\_Dept50 (Dept\_name, minimum  
salary, average\_salary, maximum\_salary) as  
select D.DNAME, MIN(E.SAL), Avg(E.SAL), MAX(E.SAL),  
from emp E, Dept D where E.Deptno = D.Deptno  
group by DNAME;

1) select minimum\_salary from emp\_Dept50  
where Dept\_name = 'SALES';

N.B: Complex view কে read only না করেও DM query  
করা সম্ভব না।

## Sequence

মাধ্যিক table র primary key মাত্র। নতুন  
data insert করলে হালে primary key থেকে  
বর্তন করলে হালে হতে। Proxy table  
বা যদি অনেক বড় data থাকে ( $1000+$ ) তখন  
নতুন data insert করলে ক্ষেত্র ব্যবহৃত primary  
key থেকে সামনে আ পরিবর্ত্য, এই issue  
solve করার জন্য sequence ক্রসক্রস রয়ে

~~create sequence student\_info~~

\* sequence র মাত্র কোন parameter নেও

মাত্র তা প্রথম হতে  $\text{next value}$  করলে minimum  
value ফিল্ট

ক্রসক্রস করলে minimum value (১)

ক্রসক্রস = remove duplicate entries

বেশি মাত্র ক্রসক্রস করলে error দেবে : ১. ১  
, ১. ১ - ১. ১ - ১. ১

MS D  $\leftrightarrow$  with student

Create sequence studentInfo\_seq

minvalue 10

maxvalue 100

Start with 20

~~NO~~

[10 is already present in the sequence]

10 with new value first

start with 20

NOCACHE

NO CYCLE

Increment by 20 ;

[Data inserted with cache]

store 100 in memory

cache use 20, 40, 60, 80, 100

nocache

$\Rightarrow$  Insert into student info

values (studentInfo\_seq.nextval,  
'nadir');

[values 100 range

cross 20, 40, 60, 80, 100

start 10 increment

repeat 20, 40, 60, 80, 100

multiple table 10

start connect 10

use cycle use

cycle 10 ]

[ 1st 10, 2nd

20, 3rd 30,

ALTER SEQUENCE studentInfo\_seq

minvalue 10

maxvalue 200

start with 20;

NO CACHE

NO CYCLE

increment by 5 ;

start with 10

minvalue 10

[ 10 ]

# ④ Sequence Drop

DROP sequence studentinfo\_seq;

Door

ofri frabut? on

Horizon 150 - oblique

## Creditos

Bogart

Digitized by srujanika@gmail.com

Adopt giftsm

1917-1920