Database Notes by Rakib

Table Of Contents

Database Part **3**
- Show Databases 3
- Create Database: 3
- Retrive Database: 3
- Update Database: 3
- Delete Database: 3

Database Table Part **4**
- Create Table: 4
- Retrive Table: 4
- Update Table: 4
- Delete Table: 4

Database Table Column **5**
- Create Column(add column): 5
- Retrive Column(Retrive Data): 5
- Update Column(Rename/Modify Name): 5
- Delete Column: 5

Database Table Row **6**
- Create Row(Insert Data): 6
- Retrive Row(Retrive Data): 6
- Update Row(Update Data): 6
- Delete Row: 6

Retrieval Queries(Select *) **7**
- Simple Select: 7
- Conditional Select: 7
- Order By Select: 7
- Distinct Select: 7
- Distinct == Group By Select: 8
- Date Select: 8
- Like Varchar Select: 8

Aggregation **10**
- Count: 10
- Min: 10

Max: 10
Group By: 10
Having: 10
Proper SQL Query: SFWGH 11

Subquery: 12
IN: 12
ANY: 12
ALL: 12

Join Operation: 13
Inner Join: joins common 13
Left Join: O><O => O> 13
Right Join: O><O => <O 13
Full Outer Join: joins all content of tables 13

Database Notes by Me

Database Part

Show Databases

-> show database;

Create Database:

-> create database "name";
-> create database seu;

Retrive Database:

-> USE 'DB_NAME';
// show all tables od the used DB
-> show tables;

Update Database:

-> CAN'T RENAME DATABASE

Delete Database:

-> drop database 'DB_NAME';

N:B: Tried to note with sequence of CRUD.
Where **CRUD** = **C**reate **R**etrive **U**pdate **D**elete.
CRUD operations are used in different applications.
Sometimes those application is called as CRUD application.

Database Table Part

Create Table:

-> create table "table name"("Attribute/Column name" variable type, Attribute name" variable type);
-> create table studentinfo(uid int(10) primary key, uname varchar(50), uaddress varchar(11), uphone int(10));

Retrive Table:

-> Desc 'Table_NAME';
// show all tables od the used DB
-> Desc studentinfo

Update Table:

-> ALTER TABLE `TableName`
RENAME TO `UpdatedTableName` ;
-> ALTER TABLE 'STUDENTINFO'
RENAME TO 'STD_INFO'

Delete Table:

-> drop table "tablename";

Database Table Column

Create Column(add column):

-> ALTER TABLE 'TABLE_NAME'
ADD 'COLUMN_NAME'   'COL_TYPE(X)';

-> ALTER TABLE std_info
ADD dept varchar(10);

Retrive Column(Retrive Data):

-> SELECT 'COLUMN_NAME' FROM 'TABLE_NAME';
-> SELECT * FROM STD_INFO;
-> SELECT DEPT, UNAME FROM STD_INFO;

Update Column(Rename/Modify Name):

// Modify Attribute Type
-> ALTER TABLE `TableName`
MODIFY 'std_no' char(8);

// Rename Column Name
ALTER TABLE `TableName`
Change Column 'std_no' 'std_no_updated' char(7);

Delete Column:

-> ALTER TABLE `TableName`
Drop Column 'ColumnName';
-> ALTER TABLE `std_info`
Drop Column 'std_no';

Database Table Row

Create Row(Insert Data):

// insert into table columns
-> insert into 'TableName' values(data, data, data, data);
-> insert into 'std_info' values(01, "Rakib", "Dhaka, Bangladesh", 019999);
-> insert into 'std_info'(uid, uname, phone) values(02, "Abdul", "Abc, Def");

Retrive Row(Retrive Data):

-> SELECT 'COLUMN_NAME' FROM 'TABLE_NAME' Where 'column_name' = 'rowData;
-> SELECT * FROM STD_INFO where uid = 01;
-> SELECT DEPT, UNAME FROM STD_INFO where uid = 01;

Update Row(Update Data):

// Modify Row Data
-> update 'Table_name' set 'Col_Name' = "Data" where 'Col_Name' = "Data";
-> update Std_Info set DEPT = "CSE" where uid = "01";
-> update Std_Info set DEPT = "CSE", uName = "Rakibul" where uid = "01";

Delete Row:

-> Delete from 'Table_name' where 'Col_Name' = "Data";
-> Delete from Std_Info where uid = "01";
-> Delete from Std_Info where uid >= "01" and uid <= "03";

Retrieval Queries(Select *)

Simple Select:

1) select * from STUDENT;
2) select STD_NO, NAME, CGPA from STUDENT;
3) select NAME, CGPA*200+200 as MARKS from STUDENT;
4) select STD_NO, NAME, upper(DEPT) from STUDENT;

Conditional Select:

11) select NAME, DEPT, CGPA from student where CGPA > 3.95;
12) select NAME, DEPT, CGPA from student where CGPA >= 3.50 && CGPA <= 3.90;
or,
select NAME, DEPT, CGPA from student where CGPA Between 3.50 and 3.90;
13) select NAME from STUDENT where DEPT = "CSE" || DEPT = "EEE";

Order By Select:

7) select DEPT from STUDENT order by dept asc;
8) select NAME from STUDENT order by GRAD_DATE asc;
select NAME from STUDENT order by NAME DESC;
9) select STD_NO, NAME, CGPA from STUDENT order by CGPA desc;
select STD_NO, NAME, CGPA from STUDENT order by NAME asc;
0) select STD_NO, NAME, CGPA from STUDENT order by CGPA desc, NAME asc;

Distinct Select:

6) select distinct DEPT from STUDENT;

Distinct == Group By Select:

6) select distinct DEPT from STUDENT;
or,
6) select DEPT from STUDENT GROUP BY DEPT;

Date Select:

14) select NAME from STUDENT where year(GRAD_DATE) = "2008" && CGPA > 3.70;
15) select * from STUDENT where year(GRAD_DATE) != "2008" && year(GRAD_DATE) != "2007";

Like Varchar Select:

16) select * from STUDENT where name like "s%"; //starts with s
-> select * from STUDENT where name like "s%a";//starts with s, ends with a
17) select * from STUDENT where NAME like "%a%a%"; // at least two 'a' in string

LIKE 'a%'	Finds any values that start with "a"
LIKE '%a'	Finds any values that end with "a"
LIKE '%abc%'	Finds any values that have "abc" in any position
LIKE '_r%'	Finds any values that have "r" in the second position

LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
LIKE 's%a'	Finds any values that start with "s" and ends with "a"

Aggregation

Count:

c) select count(std_no) from student;
or
select count(*) from student;

Min:

a)select min(cgpa) from student;

Max:

b)select max(grad_date) from student;
i) select max(cgpa) from student where year(grad_date) >= 2004;
0) select name, dept, cgpa from student where dept = "CSE" && cgpa >
(select max(cgpa) from student where dept = "EEE");

Group By:

d)select max(cgpa), min(cgpa), dept from student group by dept;
e)select max(grad_date), min(grad_date), dept from student group by dept;
f)select dept, count(std_no) from student group by dept;
g)select dept, max(cgpa), min(cgpa) from student group by dept having count(std_no) >= 3;
h) select dept, max(cgpa), min(cgpa), grad_date from student where year(grad_date) >= 2004 group by dept;

Having:

//Having is used to check info of any aggregation operation and specially after group by
-> select max(cgpa), min(cgpa), dept from student group by dept having count(*) >= 2;

Proper SQL Query: SFWGH

Select * **F**rom student **W**here id = xx and name in(select name from student where dept = xx) **G**roup by dept **H**aving count(id) >= 1;

Subquery:

a) select name from student where cgpa= (select max(cgpa) from student);

IN:

b) select name, dept, cgpa from student where (dept, cgpa) **in**(select dept, min(cgpa) from student group by dept);

ANY:

select name, dept, cgpa from student where dept = "CSE" && cgpa > **any**(select cgpa from student where dept = "EEE");

ALL:

select name, dept, cgpa from student where dept = "CSE" && cgpa > **all**(select cgpa from student where dept = "EEE");

Join Operation:

Inner Join: joins common

- select * from customerinfo c, checkinoutinfo cio where c.NID = cio.NID;
- select * from customerinfo c inner join checkinoutinfo cio on where c.NID = cio.NID;
- SELECT NAME AS COURSE_NAME, GRADE AS STUDENT_GRADE FROM GRADE G, COURSE C WHERE G.ID = C.ID AND G.STUDENT_ID = 234;
- SELECT C.NAME AS COURSE_NAME, ROW_NUMBER() OVER (ORDER BY C.NAME) AS CourseTally FROM TEACHER T, COURSE C WHERE T.ID = C.TEACHER_ID AND T.ID = 3578;

Left Join: O><O => O>

- select * from customerinfo c left join checkinoutinfo cio on c.NID = cio.NID;

Right Join: O><O => <O

- select * from customerinfo c Right join checkinoutinfo cio on c.NID = cio.NID;

Full Outer Join: joins all content of tables

- select * from customerinfo c FULL OUTER JOIN checkinoutinfo cio on c.NID = cio.NID;