

1.CREATE A TABLE

```
create table std3(  
    student_name VARCHAR(10),  
    student_id  NUMBER(10),  
    student_gender VARCHAR(10),  
    student_no NUMBER(10),  
    student_age NUMBER(10)  
);  
desc std3
```

2.INSERT :

```
insert into std3 values('sunihith','101','male','03','18');  
insert into std3 values('pavan','105','male','33','18');  
insert into std3 values('varun','104','male','32','19');  
insert into std3 values('bala','103','male','15','19');  
select * from std3
```

3.SELECT

```
select *  
from std3;
```

4.ARTHMETIC OPERATIONS

```
select student_name,student_age+3  
from std3
```

5.ALIAS

```
select student_name,student_id as "REGISTER_NO"  
from std3
```

6. CONCATINATION

```
select student_name,student_age||'    '||student_id as "REGISTER_NO"  
from std3
```

7.DESCRIBE

```
create table std3(  
    student_name VARCHAR(10),  
    student_id  NUMBER(10),  
    student_gender VARCHAR(10),  
    student_no NUMBER(10),  
    student_age NUMBER(10)  
);  
desc std3
```

8.WHERE

```
select student_name,student_age  
from std3  
where student_age= 18;
```

9.BETWEEN AND

```
select student_name,student_id,student_gender  
from std3  
where student_id BETWEEN 102 AND 104;
```

10.IN OPERATOR

```
select student_name,student_id,student_gender
from std3
where student_id IN (101,102,103);
```

11.NOT IN OPERATOR

```
select student_name,student_id,student_gender
from std3
where student_id NOT IN (101,103);
```

12.LIKE OPERATOR

```
select student_name,student_id,student_gender
from std3
where student_name LIKE 's%';
```

13.NULL

A)IS NULL

```
select student_name,student_id,student_gender
from std3
where student_id IS NULL ;
```

B)IS NOT NULL

```
select student_name,student_id,student_gender
from std3
where student_id IS NOT NULL ;
```

14.LOGICAL OPERATORS

A)AND

```
select student_name,student_id,student_gender  
from std3  
where student_id >=101 AND student_name LIKE 's%' ;
```

B)OR

```
select student_name,student_id,student_gender  
from std3  
where student_id >=101 OR student_name LIKE 's%' ;
```

15.ORDER BY(BY DEFAULT ASC)

```
select student_name,student_id,student_gender  
from std3  
ORDER BY student_id ;
```

A)ASC

```
select student_name,student_id,student_gender  
from std3  
ORDER BY student_id ASC ;
```

B)DESC

```
select student_name,student_id,student_gender  
from std3  
ORDER BY student_id DESC;
```

16. Change data type in table

```
select * from mor
alter table mor
add (m_time_temp number(20,2));
update mor set m_time_temp=m_time
alter table mor drop column m_time
alter table mor rename column m_time_temp to m_time
alter table mor rename column m_time to m_time_temp
```

17. update name

```
update reddy
set f_name='reddys'
where spent_m='123'
select *
from reddy
```

18. delete column

```
select *
from reddy
delete from reddy where spent_m='1236'
```

19. select top clause

```
select * from(
    select * from reddy
)where ROWNUM<=3
```

20. fetch the number of rows

```
select f_name  
from reddy  
order by f_name  
fetch first 3 rows only
```

21. Fetch first 50 rows only

```
fetch first 3 rows only  
select * from reddy fetch first 50 percent rows only
```

22. top 3 In order by

```
select * from(  
    select * from reddy  
    order by age desc  
)where ROWNUM<=3
```

23. Aggregate Functions

A) Count

```
SELECT COUNT(*) AS total_rows FROM reddy;
```

B) Sum

```
SELECT SUM(spent_m) AS total_spent FROM reddy;
```

C) Average

```
SELECT AVG(age) AS average_age FROM reddy;
```

D) Min

```
SELECT MIN(age) AS youngest_age FROM reddy;
```

E) Max

```
SELECT MAX(age) AS oldest_age FROM reddy;
```

23. Listagg

```
FROM reddy;  
SELECT LISTAGG(f_name, ', ') WITHIN GROUP (ORDER BY f_name) AS names  
FROM reddy;
```

24. Using Aggregate Functions with GROUP BY

```
SELECT what_d, AVG(spent_m) AS avg_spent  
FROM reddy  
GROUP BY what_d;
```

25. af with where

```
SELECT COUNT(*) AS count_over_5  
FROM reddy  
WHERE age > 5;
```

26. af with count

```
);  
SELECT COUNT(*) AS total_rows FROM reddy;
```

27. af with having

```
select what_d,sum(spent_m) as total_spent
from reddy
group by what_d
having sum(spent_m)>1235
```

28. adding new column to old records default value

```
alter table reddy
add brand varchar(10)
update reddy
set brand='hp'
select * from reddy
```

29. adding new column to old records different values

```
update reddy
set brand=case
when what_d='farmer' then 'hp'
when what_d='housewife' then 'dell'
when what_d='student' then 'asus'
when what_d='nothing' then 'victus'
else 'other'
end;
select * from reddy
```

30. count by where


```
select count(age)
from reddy where age>5
```

31. count by distinct

```
select count(distinct age)
from reddy
```

32. count (*) distinct

```
select count(*) as "number of records"
from reddy
```

33. Use COUNT() with GROUP BY

```
select count(*) as "number of records", age
from reddy
group by age
```

34. like using where

```
SELECT *
FROM reddy
WHERE what_d LIKE '%student%';
```

35. wild card charecters

```
SELECT *  
FROM reddy  
WHERE what_d LIKE '%student%';
```

```
SELECT *  
FROM reddy  
WHERE what_d LIKE 'stu%';
```

```
SELECT *  
FROM reddy  
WHERE what_d LIKE '_s%';
```

```
SELECT *  
FROM reddy  
WHERE what_d LIKE '_____';
```

36. between in

```
SELECT *  
FROM reddy  
WHERE age BETWEEN 3 AND 6;
```

A) In

```
SELECT *  
FROM reddy  
WHERE what_d IN ('student', 'farmer', 'housewife');
```

37. between and , in

```
SELECT *  
FROM reddy  
WHERE age BETWEEN 3 AND 6  
AND what_d IN ('student', 'farmer');
```

38. Concatenate

```
SELECT f_name || ' ' || l_name AS full_name,  
       age,  
       what_d,  
       spent_m  
FROM reddy;
```

39.AS

```
9  
0  SELECT spent_m AS amount_spent  
1  FROM reddy;  
2
```

40.use column as

```
SELECT f_name,  
       l_name,  
       age * 2 AS double_age  
FROM reddy;
```

41.as with on

```
SELECT a.f_name, a.l_name, b.product_name  
FROM reddy a  
JOIN orders b ON a.f_name = b.customer_name;
```

