

## SECTION 4

### TRUNC

select hiredate,

trunc(hiredate,'month')

from singer;

Results		Explain	Describe	Saved SQL	History
HIREDATE		TRUNC(HIREDATE,'MONTH')			
10-Mar-2024		01-Mar-2024			
05-Sep-2024		01-Sep-2024			
01-Oct-2024		01-Oct-2024			
09-Oct-2024		01-Oct-2024			
14-Dec-2024		01-Dec-2024			
5 rows returned in 0.01 seconds					
<a href="#">Download</a>					

### ROUND TO HIREDATE

select hiredate,

round(hiredate,'month')

from singer;

Results

Explain

Describe

Saved SQL

History

HIREDATE	ROUND(HIREDATE,'MONTH')
10-Mar-2024	01-Mar-2024
05-Sep-2024	01-Sep-2024
01-Oct-2024	01-Oct-2024
09-Oct-2024	01-Oct-2024
14-Dec-2024	01-Dec-2024

5 rows returned in 0.01 seconds

Download

### LAST DAY

```
select last_day(sysdate)
```

```
from dual;
```

Results	Explain	Describe	Saved SQL	History
LAST_DAY(SYSDATE)				
31-Aug-2024				
1 rows returned in 0.00 seconds <a href="#">Download</a>				

### NEXT DAY

```
select next_day(sysdate,'monday')
```

```
from dual;
```

Results	Explain	Describe	Saved SQL	History
NEXT_DAY(SYSDATE,'MONDAY')				
12-Aug-2024				
1 rows returned in 0.00 seconds <a href="#">Download</a>				

### ADD MONTHS

```
select add_months(sysdate,5)
```

```
from dual;
```

Results	Explain	Describe	Saved SQL	History
ADD_MONTHS(SYSDATE,5)				
05-Jan-2025				
1 rows returned in 0.01 seconds <a href="#">Download</a>				

### MONTHS BETWEEN

```
select id,name
```

from singer

where months\_between

(sysdate,hiredate)<100;

Results	Explain   Describe   Saved SQL   History
ID	NAME
100	rahul
102	vijay
103	vishnu
101	harsha
104	harshini
5 rows returned in 0.00 seconds <a href="#">Download</a>	

## SECTION 5

### NVL FUNCTION

select id,nvl(dept,'not assigned')

from singer;

Results

Explain

Describe

Saved SQL

History

ID	NVL(DEPT,'NOTASSIGNED')
100	not assigned
102	violinist
103	tabla
101	not assigned
104	flute

5 rows returned in 0.01 seconds

Download

### NVL DATE

```
SELECT NVL(TO_CHAR(hiredate, 'YYYY-MM-DD'), 'no date')
FROM singer;
```

NVL(TO_CHAR(HIREDATE,'YYYY-MM-DD'),'NODATE')
2024-03-10
2024-09-05
2024-10-01
2024-10-09
2024-12-14
5 rows returned in 0.00 seconds <a href="#">Download</a>

### CHARACTER TO DATE

```
select to_date('may10,1989','fmondd,yyyy') as "convert"
from dual;
```

Results	Explain	Describe	Saved SQL	History
convert				
10-May-1989				
1 rows returned in 0.00 seconds <a href="#">Download</a>				

### NUMBER TO CHARACTER

```
select to_char(salary,'$99,999')
from singer;
```

Results	Explain	Describe	Saved SQL	History
TO_CHAR(SALARY,'\$99,999')				
\$15,000				
\$20,000				
\$20,000				
\$20,000				
\$10,000				

### DATE TO CHARACTER

```
select to_char(hiredate,'month dd,yyyy')
from singer;
```

Results	Explain	Describe	Saved SQL	History
TO_CHAR(HIREDATE,'MONTHDD,YYYY')				
march 10,2024				
september 05,2024				
october 01,2024				
october 09,2024				
december 14,2024				
5 rows returned in 0.00 seconds <a href="#">Download</a>				

### DATE FUNCTION

```
select id,name,hiredate,round(months_between(sysdate,hiredate)) as tenure,
add_months(hiredate,6) as review, next_day(hiredate,'friday'),last_day(hiredate)
from singer;
```

Results	Explain	Describe	Saved SQL	History		
ID	NAME	HIREDATE	TENURE	REVIEW	NEXT_DAY(HIREDATE,'FRIDAY')	LAST_DAY(HIREDATE)
100	rahul	10-Mar-2024	5	10-Sep-2024	15-Mar-2024	31-Mar-2024
102	vijay	05-Sep-2024	-1	05-Mar-2025	06-Sep-2024	30-Sep-2024
103	vishnu	01-Oct-2024	-2	01-Apr-2025	04-Oct-2024	31-Oct-2024
101	harsha	09-Oct-2024	-2	09-Apr-2025	11-Oct-2024	31-Oct-2024

## SECTION 6

### HEARIACHEY LEVEL

select level id,name,deptid

from emp

start with id=100

connect by prior id=deptid;

Results	Explain	Describe	Saved SQL	History
ID	NAME	DEPTID		
1	rahul	12		

1 rows returned in 0.00 seconds [Download](#)

### HIERACHIAL USING START WITH KEYWORD

select id,name,deptid

from emp

start with id=100

connect by prior id=deptid;

Results	Explain	Describe	Saved SQL	History
ID	NAME	DEPTID		
1	rahul	12		

1 rows returned in 0.00 seconds [Download](#)

### FULL OUTER JOIN

select e.id,e.name,d.deptid,d.dept\_name

```
from emp e full outer join dept d
on (e.deptid=d.deptid);
```

Results Explain Describe Saved SQL History			
ID	NAME	DEPTID	DEPT_NAME
103	harsha	13	physics
104	harshitha	13	physics
105	harshini	14	maths
100	rahul	12	chemistry
102	vijay	13	physics
-	-	15	bioo
6 rows returned in 0.01 seconds <a href="#">Download</a>			

## RIGHT OUTER JOIN

```
select e.id,e.name,d.deptid,d.dept_name
from emp e right outer join dept d
on (e.deptid=d.deptid);
```

Results Explain Describe Saved SQL History			
ID	NAME	DEPTID	DEPT_NAME
103	harsha	13	physics
104	harshitha	13	physics
105	harshini	14	maths
100	rahul	12	chemistry
102	vijay	13	physics
-	-	15	bioo
6 rows returned in 0.01 seconds <a href="#">Download</a>			

## LEFT OUTER JOIN

```
select e.id,e.name,d.deptid,d.dept_name
from emp e left outer join dept d
on (e.deptid=d.deptid);
```

Results	Explain	Describe	Saved SQL	History
ID	NAME	DEPTID	DEPT_NAME	
103	harsha	13	physics	
104	harshitha	13	physics	
105	harshini	14	maths	
100	rahul	12	chemistry	
102	vijay	13	physics	
5 rows returned in 0.00 seconds <a href="#">Download</a>				

## ON CLAUSE

```
select id,name,dept_name
```

```
from emp e join dept d
```

```
on(e.deptid=d.deptid);
```

Results	Explain	Describe	Saved SQL	History
ID	NAME	DEPT_NAME		
103	harsha	physics		
104	harshitha	physics		
105	harshini	maths		
100	rahul	chemistry		
102	vijay	physics		
5 rows returned in 0.01 seconds <a href="#">Download</a>				

## USING CLAUSE

```
select id,name,deptid,dept_name
```

```
from emp join dept using (deptid);
```



Results

Explain

Describe

Saved SQL

History

ID	NAME	DEPT_NAME
103	harsha	physics
104	harshitha	physics
105	harshini	maths
100	rahul	chemistry
102	vijay	physics

5 rows returned in 0.01 seconds

Download

## CROSS JOIN

```
select id,name,dept_name
```

```
from emp cross join dept;
```

Results	Explain	Describe	Saved SQL	History
ID	NAME	DEPT_NAME		
103	harsha	maths		
104	harshitha	maths		
105	harshini	maths		
100	rahul	maths		
102	vijay	maths		
103	harsha	physics		
104	harshitha	physics		

## NATURAL JOIN

```
select id,name,deptid,dept_name
```

```
from emp natural join dept;
```

Results	Explain	Describe	Saved SQL	History
ID	NAME	DEPTID	DEPT_NAME	
103	harsha	13	physics	
104	harshitha	13	physics	
105	harshini	14	maths	
100	rahul	12	chemistry	
102	vijay	13	physics	
5 rows returned in 0.00 seconds <a href="#">Download</a>				

## SECTION 7

### EQUIJOIN AND CARTESIAN PRODUCT

```
CREATE TABLE employ(
  eno VARCHAR(14),
  ename VARCHAR(14),
  eadhress VARCHAR(15),
  epno VARCHAR(15),
  depno VARCHAR(14),
  depname VARCHAR(14),
  jobid VARCHAR(10),
  salary VARCHAR(10),
  create_date DATE DEFAULT SYSDATE);
```

ENO	ENAME	EADHRESS	EPNO	DEPNO	DEPNAME	JOBID	SALARY	CREATE_DATE
03	mahat	chennai	684	3456433	ece	5698	700000	26-Jul-2024
01	deepa	tpt	3256	3456433	cse	9954	2568752	26-Jul-2024
04	mahath	chennai	5564	3456433	it	22313	3300000	26-Jul-2024
05	mahi	chenai	7523	3456433	aii	68876	3695000	26-Jul-2024

```
CREATE TABLE jobs (
  job_id VARCHAR(10) PRIMARY KEY,
  job_title VARCHAR(50) NOT NULL,
  min_salary DECIMAL(8, 2),
```

max\_salary DECIMAL(8, 2)

);

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
22313	CEO	100000	300000
9954	manager	45000	90000
5698	software	50000	70000

## PROPRIETARY JOINS

SELECT employ.ename, jobs.job\_title

FROM employ,jobs

WHERE employ.jobid=jobs.job\_id;

ENAME	JOB_TITLE
mahat	software
deepa	manager
mahath	CEO

3 rows returned in 0.00 seconds [Download](#)

## EQUIJOIN

SELECT employ.ename, employ.jobid, jobs.job\_title

FROM employ, jobs

WHERE employ.jobid = jobs.job\_id;

Results	Explain	Describe	Saved SQL	History
ENAME	JOBID	JOB_TITLE		
mahat	5698	software		
deepa	9954	manager		
mahath	22313	CEO		

3 rows returned in 0.01 seconds [Download](#)

## ALIASES

SELECT ename, e.jobid, job\_title

FROM employ e, jobs j

WHERE e.jobid = j.job\_id

AND depno=3456433;

Results		
<a href="#">Results</a> <a href="#">Explain</a> <a href="#">Describe</a> <a href="#">Saved SQL</a> <a href="#">History</a>		
ENAME	JOBID	JOB_TITLE
mahat	5698	software
deepa	9954	manager
mahath	22313	CEO
3 rows returned in 0.00 seconds <a href="#">Download</a>		

## CARTESIAN PRODUCT JOIN

```
SELECT employ.ename,jobs.job_title
FROM employ,jobs;
```

ENAME	JOB_TITLE
mahat	CEO
deepa	CEO
mahath	CEO
mahi	CEO
mahat	manager
deepa	manager
mahath	manager
mahi	manager
mahat	software
deepa	software
mahath	software
mahi	software

## JOIN

```
SELECT last_name, location
FROM Employees e, departments d, jobs j
WHERE e.job_id = j.job_id
AND e.department_id =d.department_id;
```

Results	
<a href="#">Results</a> <a href="#">Explain</a> <a href="#">Describe</a> <a href="#">Saved SQL</a> <a href="#">History</a>	
LAST_NAME	LOCATION
Doe	Chicago
1 rows returned in 0.01 seconds <a href="#">Download</a>	

## NON EQUIJOIN

```
SELECT ename,salary,grade,lowсал,
highsal
FROM employ,job_grade
WHERE (salary BETWEEN lowсал AND highсал);
```

Results	Explain	Describe	Saved SQL	History
ENAME	SALARY	GRADE	LOWSAL	HIGHSAL
mahat	700000	B	30000	80000
mahi	3695000	D	3500001	4000000
mahi	3695000	B	30000	80000
mahath	3300000	D	3000001	3500000
mahath	3300000	B	30000	80000
deepa	2568752	A	1000	2999

6 rows returned in 0.00 seconds [Download](#)

## SECTION 8

### MAX

```
select max(salary)
from singer;
```

Results	Explain	Describe	Saved SQL	History
MAX(SALARY)				
20000				
1 rows returned in 0.00 seconds		<a href="#">Download</a>		

### MIN

```
select min(name)
from singer;
```

Results	Explain	Describe	Saved SQL	History
MIN(NAME)				
harsha				
1 rows returned in 0.00 seconds <a href="#">Download</a>				

## SUM

```
select sum(salary)
from singer;
```

Results	Explain	Describe	Saved SQL	History
SUM(SALARY)				
85000				
1 rows returned in 0.00 seconds <a href="#">Download</a>				

## VARIENCE

```
select round(variance(salary),4)
from singer;
```

Results	Explain	Describe	Saved SQL	History
ROUND(VARIANCE(SALARY),4)				
20000000				
1 rows returned in 0.00 seconds <a href="#">Download</a>				

## COUNT

```
select count(id)
```

```
from singer;
```

Results	Explain	Describe	Saved SQL	History
COUNT(ID)				
5				
1 rows returned in 0.00 seconds <a href="#">Download</a>				

## GROUP FUNCTION

```
select max(salary),min(salary),min(hiredate)
```

```
from singer;
```

Results

Explain

Describe

Saved SQL

History

MAX(SALARY)	MIN(SALARY)	MIN(HIREDATE)
20000	10000	10-Mar-2024

1 rows returned in 0.00 seconds

Download

## DEFAULT

```
create table my_emp(
```

```
hiredate date default sysdate,  
name varchar(20),  
);
```

## SECTION 9

### GROUP BY

```
SELECT id, AVG(salary)  
FROM singer  
GROUP BY id  
ORDER BY id;
```

Results	Explain	Describe	Saved SQL	History
ID	AVG(SALARY)			
100	15000			
101	20000			
102	20000			
103	20000			
104	10000			
5 rows returned in 0.00 seconds			<a href="#">Download</a>	

### EXAMPLE

```
SELECT MAX(salary)  
FROM singer  
GROUP BY id;
```



Results	Explain	Describe	Saved SQL	History
MAX(SALARY)				
15000				
20000				
20000				
20000				
10000				
5 rows returned in 0.00 seconds <a href="#">Download</a>				

## COUNT

SELECT COUNT(name), id

FROM singer

GROUP BY id

ORDER BY id;

Results	Explain	Describe	Saved SQL	History
COUNT(NAME)				
ID				
1				
100				
1				
101				
1				
102				
1				
103				
1				
104				
5 rows returned in 0.00 seconds <a href="#">Download</a>				

## WHERE CLAUSE

SELECT id, MAX(salary)

FROM singer

WHERE name != 'harsha'

GROUP BY id;

Results	Explain	Describe	Saved SQL	History
ID	MAX(SALARY)			
100	15000			
102	20000			
103	20000			
104	10000			
4 rows returned in 0.01 seconds			<a href="#">Download</a>	

### MORE GROUP BY

SELECT id, ROUND(AVG(salary)) AS salary

FROM singer

GROUP BY id

ORDER BY id;

Results	Explain	Describe	Saved SQL	History
ID	SALARY			
100	15000			
101	20000			
102	20000			
103	20000			
104	10000			
5 rows returned in 0.01 seconds			<a href="#">Download</a>	

### NESTING GROUP FUNCTIONS

SELECT max(avg(salary))


FROM singer

GROUP by id;

Results	Explain	Describe	Saved SQL	History
MAX(AVG(SALARY))				
20000				
1 rows returned in 0.01 seconds <a href="#">Download</a>				

## HAVING

```
SELECT id, MAX(salary)
FROM singer
WHERE COUNT(*) > 1
GROUP BY id;
```

Results	Explain	Describe	Saved SQL	History
 Error at line 3/7: ORA-00934: group function is not allowed here				

## UNION

```
SELECT id
FROM singer
UNION
SELECT id
FROM emp;
```

Results	Explain	Describe	Saved SQL	History
ID				
100				
101				
102				
103				
104				
105				

### UNION ALL

```
SELECT id
FROM singer
UNION ALL
SELECT id
FROM emp;
```

Results	Explain	Describe	Saved SQL	History
ID				
100				
101				
102				
103				
104				
100				
102				
103				
104				
105				

## INTERSECT

```
SELECT id
FROM singer
INTERSECT
SELECT id
FROM emp;
```

Results	Explain	Describe	Saved SQL	History
ID				
100				
102				
103				
104				
4 rows returned in 0.00 seconds				
<a href="#">Download</a>				

## MINUS

```
SELECT id  
FROM singer  
MINUS  
SELECT id  
FROM emp;
```

Results	Explain	Describe	Saved SQL	History
ID				
101				
1 rows returned in 0.01 seconds <a href="#">Download</a>				

## SECTION 10

### INTERSECT:

```
SELECT emp_no  
FROM employee  
intersect  
SELECT job_id  
FROM job;
```

Results	Explain	Describe	Saved SQL	History
no data found				

### MINUS:

```
SELECT emp_no  
FROM employee  
minus  
SELECT job_id
```

FROM job;

EMP_NO
101
105
108
123
145

#### SUBQUERY EXAMPLE:

```
SELECT e_name,  
emp_no  
FROM employee  
WHERE emp_no >  
(SELECT emp_no  
FROM employee  
WHERE e_name = 'lucky');
```

E_NAME	EMP_NO
ram	145

1 rows returned in 0.00 seconds [Download](#)

#### SUBQUERY AND NULL:

```
SELECT e_name,  
emp_no  
FROM employee  
WHERE emp_no >  
(SELECT emp_no  
FROM employee
```

```
WHERE e_name = 'anji');
```

Results	Explain	Describe	Saved SQL	History
no data found				

#### SUBQUERY FROM DIFFERENT TABLES:

```
SELECT e_name, job_id, dept_no  
FROM employee  
WHERE job_id =  
(SELECT job_id  
FROM job  
WHERE job_name = 'teacher')  
ORDER BY job_id;
```

E_NAME	JOB_ID	DEPT_NO
ragni	9	13
ram	9	-
rows returned in 0.00 seconds <a href="#">Download</a>		

#### GROUP FUNCTIONS IN SUBQUERIES:

```
SELECT e_name, salary  
FROM employee  
WHERE salary <  
(SELECT AVG(salary)
```



FROM employee);

E_NAME	SALARY
ram	45678
raju	7890
lucky	12345
rani	70890
4 rows returned in 0.00 seconds <a href="#">Download</a>	

### CONCATENATION:

```
select dept_no||' '||dept_name  
from employee;
```

DEPT_NO  ' '  DEPT_NAME
telugu
23maths
45math
13science
13social
5 rows returned in 0.00 seconds <a href="#">Download</a>

## SECTION 12

### CREATE COPY OF TABLE

```
CREATE TABLE copy_singer
```

```
AS (SELECT * FROM singer);
```

Results

Explain

Describe

Saved SQL

History

Table created.

0.03 seconds

## SYNTAX TO CREATE A COPY OF TABLE

DESc copy\_singer;

Results	Explain	Describe	Saved SQL	History					
Object Type		TABLE ?	Object	COPY_SINGER ?					
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COPY_SINGER	ID	NUMBER	-	15	0	-	✓	-	-
	NAME	VARCHAR2	20	-	-	-	✓	-	-
	DEPT	VARCHAR2	20	-	-	-	✓	-	-
	HIREDATE	DATE	7	-	-	-	✓	-	-
	SALARY	NUMBER	-	15	0	-	✓	-	-

SELECT \* FROM copy\_singer;

Results	Explain	Describe	Saved SQL	History
ID	NAME	DEPT	HIREDATE	SALARY
100	rahul	-	10-Mar-2024	15000
102	vijay	violinist	05-Sep-2024	20000
103	vishnu	tabla	01-Oct-2024	20000
101	harsha	-	09-Oct-2024	20000
104	harshini	flute	14-Dec-2024	10000

5 rows returned in 0.00 seconds
Download

## INSERT

INSERT INTO copy\_singer

VALUES (200,'Hanuman', 'singer','05-sep-2005', 15000);

Results	Explain	Describe	Saved SQL	History
1 row(s) inserted.				
0.00 seconds				

### INSERTING WITH NOT ENOUGH VALUES

INSERT INTO copy\_singer

(id, name, dept, hiredate,salary)

VALUES

(302,'Grigorz','Polanski',4200);

Results	Explain	Describe	Saved SQL	History
Error at line 3/1: ORA-00947: not enough values ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 828 ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658 ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 813 ORA-06512: at "APEX_220200.WWV_FLOW_DYNAMIC_EXEC", line 2046  1. INSERT INTO copy_singer 2. (id, name, dept, hiredate,salary) 3. VALUES 4. (302,'Grigorz','Polanski',4200);				

### INSERTING WITH NULL VALUES

INSERT INTO copy\_singer

(id, name, dept, hiredate,salary)

VALUES

(302,'Grigorz','Polanski','',4200);

Results	Explain	Describe	Saved SQL	History
1 row(s) inserted.				
0.01 seconds				

### INSERTING SEPECIAL VALUES

INSERT INTO copy\_singer

(id, name, dept, hiredate,salary)

VALUES

(302,'Grigorz','Polanski',sysdate,4200);

Results	Explain	Describe	Saved SQL	History
1 row(s) inserted.				
0.01 seconds				

### INSERTING SPECIFIC DATE VALUES

SELECT name, TO\_CHAR(hiredate,'Month, fmdd, yyyy')

FROM singer

WHERE id = 101;

Results	Explain	Describe	Saved SQL	History
NAME	TO_CHAR(HIREDATE,'MONTH,FMDD,YYYY')			
harsha	October , 9, 2024			
1 rows returned in 0.01 seconds				
<a href="#">Download</a>				

### UPDATE

UPDATE copy\_singer

```
SET hiredate = sysdate
```

```
WHERE id = 101;
```

Results	Explain	Describe	Saved SQL	History
1 row(s) updated.				
0.00 seconds				

### UPDATING A COLUMN

```
UPDATE copy_singer
```

```
SET salary = (SELECT salary
```

```
FROM copy_singer
```

```
WHERE id = 100)
```

```
WHERE id = 101;
```

Results	Explain	Describe	Saved SQL	History
1 row(s) updated.				
0.01 seconds				

### UPDATING 2 COLOUMS

```
UPDATE copy_singer
```

```
SET salary = (SELECT salary
```

```
FROM copy_singer
```

```
WHERE id = 103),
```

```
hiredate = (SELECT hiredate
```

```
FROM copy_singer
```

```
WHERE id = 103)
```

```
WHERE id = 104;
```

Results	Explain	Describe	Saved SQL	History
<div>Saved SQL</div> <p>1 row(s) updated.</p> <p>0.01 seconds</p>				

### UPDATING ROWS BASED ON ANOTHER COLUMN

```
UPDATE copy_singer
SET salary = (SELECT salary
FROM singer
WHERE id = 100)
WHERE id = 103;
```

Results	Explain	Describe	Saved SQL	History
<p>1 row(s) updated.</p> <p>0.00 seconds</p>				

### DELETE

```
DELETE from copy_singer
WHERE id = 103;
```

Results	Explain	Describe	Saved SQL	History
<p>1 row(s) deleted.</p> <p>0.00 seconds</p>				

### SUBQUERY DELETE

```
DELETE FROM copy_singer
WHERE id =
(SELECT id
FROM singer
WHERE dept = 'flute');
```

Results	Explain	Describe	Saved SQL	History
1 row(s) deleted.				
0.00 seconds				

## SECTION 13

### DATA DICTONARY

```
SELECT table_name, status
FROM USER_TABLES;
```

Results	Explain	Describe	Saved SQL	History
TABLE_NAME		STATUS		
CLIENTS		VALID		
COPY_JOB_HISTORY		VALID		
COURSE		VALID		
DEPARTMENT		VALID		
DEPT		VALID		
DOC		VALID		
DOCTOR		VALID		
EMP		VALID		
EMPLOYEE		VALID		
FACULTY		VALID		

### CREATE TABLE MY FRIENDS:

```
CREATE TABLE my_friends  
(first_name VARCHAR2(20),  
  
last_name VARCHAR2(30),  
email VARCHAR2(30),  
phone_num VARCHAR2(12),  
birth_date DATE);
```

### CREATE TABLE MY CDCOLLECTION:

```
CREATE TABLE my_cd_collection  
(cd_number NUMBER(3),  
title VARCHAR2(20),  
artist VARCHAR2(20),  
purchase_date DATE DEFAULT SYSDATE);
```

Results	Explain	Describe	Saved SQL	History
Table created.				
0.02 seconds				

### DATA DICTIONARY:

```
SELECT table_name, status  
FROM USER_TABLES;
```



Results	Explain	Describe	Saved SQL	History
TABLE_NAME		STATUS		
COPY_EMPLOYEE		VALID		
COPY_JOB		VALID		
COURSE		VALID		
DEPARTMENT		VALID		
EMPLOYEE		VALID		
FACULTY		VALID		
HTMLDB_PLAN_TABLE		VALID		
JOB		VALID		
MY_CD_COLLECTION		VALID		

#### DATA DICTIONARY TABLE NAME:

SELECT table\_name, status

FROM ALL\_TABLES;

Results	Explain	Describe	Saved SQL	History
TABLE_NAME		STATUS		
COPY_EMPLOYEE		VALID		
COPY_JOB		VALID		
COURSE		VALID		
DEPARTMENT		VALID		
EMPLOYEE		VALID		
FACULTY		VALID		
HTMLDB_PLAN_TABLE		VALID		
JOB		VALID		
MY_CD_COLLECTION		VALID		

#### TIMES STAMP TABLE:

CREATE TABLE time\_ex1

```
(exact_time TIMESTAMP);
```

#### TIMES STAMP TABLE INSERT ROW:

```
INSERT INTO time_ex1
```

```
VALUES ('8-aug-2024 10:52:29.123456');
```

```
1 row(s) inserted.
```

```
0.00 seconds
```

#### TIMES STAMP TABLE INSERT ROW USING SYS UPDATE:

```
INSERT INTO time_ex1
```

```
VALUES (SYSDATE);
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

```
1 row(s) inserted.
```

```
0.00 seconds
```

#### SYS UPDATE:

```
INSERT INTO time_ex1
```

```
VALUES (SYSTIMESTAMP);
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

```
1 row(s) inserted.
```

```
0.00 seconds
```

#### DISPLAY TIME STAMP:

```
SELECT *
```

FROM time\_ex1;

EXACT_TIME
08-AUG-24 10.52.29.123456 AM
05-AUG-24 04.48.13.000000 AM
08-AUG-24 10.52.29.123456 AM
07-AUG-24 03.53.52.000000 AM
07-AUG-24 03.54.41.711002 AM
05-AUG-24 04.48.41.723036 AM
6 rows returned in 0.00 seconds <a href="#">Download</a>

#### CREATE TABLE:

```
CREATE TABLE time_ex4
```

```
(loan_duration1 INTERVAL YEAR(3) TO MONTH,
```

```
loan_duration2 INTERVAL YEAR(2) TO MONTH);
```

Results	Explain	Describe	Saved SQL	History
Table created.				

#### INSERT :

```
INSERT INTO time_1(loan_duration1, loan_duration2)
```

```
VALUES (INTERVAL '120' MONTH(3),
```

```
INTERVAL '3-6' YEAR TO MONTH);
```

Results	Explain	Describe	Saved SQL	History
1 row(s) inserted.				

INTERVAL MONTH TO YEAR:

```
SELECT SYSDATE + loan_duration1 AS "120 months from now",
SYSDATE + loan_duration2 AS "3 years 6 months from
now"
FROM time_1;
```

120 months from now	3 years 6 months from now
07-Aug-2034	07-Feb-2028
1 rows returned in 0.01 seconds <a href="#">Download</a>	

**DAY TO SECOND:**

```
CREATE TABLE time_2
(day_duration1 INTERVAL DAY(3) TO SECOND,
day_duration2 INTERVAL DAY(3) TO SECOND);
```

Results	Explain	Describe	Saved SQL	History
Table created.				

**DAY TO SECOND INSERT ROW:**

```
INSERT INTO time_2 (day_duration1, day_duration2)
VALUES (INTERVAL '25' DAY(2), INTERVAL '4 10:30:10' DAY TO
SECOND);
```

1 row(s) inserted.

0.00 seconds

#### DAY TO SECOND SELECT:

```
SELECT SYSDATE + day_duration1 AS "25 Days from now",  
TO_CHAR(SYSDATE + day_duration2, 'dd-Mon-yyyy hh:mi:ss')  
AS "precise days and time from now"  
FROM time_2;
```

Results	Explain	Describe	Saved SQL	History
25 Days from now		precise days and time from now		
01-Sep-2024		11-Aug-2024 02:58:09		
01-Sep-2024		11-Aug-2024 02:58:09		
01-Sep-2024		11-Aug-2024 02:58:09		
5 rows returned in 0.01 seconds		<a href="#">Download</a>		

## SECTION 14

### CREATING TABLE

```
CREATE TABLE clients (  
client_number NUMBER(4),  
first_name VARCHAR2(14),  
last_name VARCHAR2(13));
```

### CREATING CONSTRAINTS

```
CREATE TABLE clients  
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk  
PRIMARY KEY,
```

```
first_name VARCHAR2(14),  
last_name VARCHAR2(13));
```

### **COLUMN LEVEL**

```
CREATE TABLE clients  
(client_number NUMBER(4) CONSTRAINT clients_cient_num_pk PRIMARY KEY,  
last_name VARCHAR2(13) CONSTRAINT clients_last_name_nn NOT NULL,  
email VARCHAR2(80) CONSTRAINT clients_email_uk UNIQUE);
```

### **TABLE LEVEL**

```
CREATE TABLE clients (  
    client_number NUMBER(6) NOT NULL,  
    first_name VARCHAR2(20),  
    last_name VARCHAR2(20),  
    phone VARCHAR2(20),  
    email VARCHAR2(10) NOT NULL,  
    CONSTRAINT clients_phone_email_uk UNIQUE (email,phone));  
)
```

### **VIOLATION**

```
CREATE TABLE clients(  
    client_number NUMBER(6),  
    first_name VARCHAR2(20),  
    last_name VARCHAR2(20),  
    phone VARCHAR2(20) CONSTRAINT phone_email_uk  
    UNIQUE(email,phone),  
    email VARCHAR2(10) CONSTRAINT NOT NULL,  
    CONSTRAINT emailclients_email NOT NULL,  
    CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```

### **UNIQUE CONSTRAINT**

```
INSERT INTO clients (client_number, first_name, Last_name, phone,  
email)  
VALUES (7234, 'Lonny', 'Vigil', 4072220091, 'lbv@lbv.net');
```

### **PRIMARY KEY**

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk
PRIMARY KEY,
first_name VARCHAR2(14),
last_name VARCHAR2(13));
```

#### **TABLE LEVAL**

```
CREATE TABLE copy_job_history
(employee_id NUMBER(6,0),
start_date DATE,
job_id VARCHAR2(10),
department_id NUMBER(4,0),
CONSTRAINT copy_jhist_id_st_date_pk PRIMARY KEY(employee_id,
start_date));
```

#### **FOREIGN KEY**

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
first_name VARCHAR2(20),
last_name VARCHAR2(25),
department_id NUMBER(4,0),
email VARCHAR2(25),
CONSTRAINT c_emps_dept_id_fk FOREIGN KEY (department_id)
REFERENCES departments(department_id));
```

#### **ON DELETE CASCADE**

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
first_name VARCHAR2(20),
last_name VARCHAR2(25),
department_id NUMBER(4,0),
email VARCHAR2(25),
CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
REFERENCES copy_departments(department_id));
```

## SECTION 15

### VIEW:

```
CREATE VIEW view_employee  
AS SELECT emp_no,e_name,job_id  
FROM employee  
WHERE emp_no BETWEEN 100 and 243569;
```

View created.

0.01 seconds

### VIEWING THE TABLE:

```
SELECT *  
FROM view_employee;
```

EMP_NO	E_NAME	JOB_ID
101	raju	1
105	rani	5
108	ragni	9
123	lucky	1
145	ram	9

5 rows returned in 0.01 seconds

[Download](#)

### CREATING A VIEW:

```
CREATE OR REPLACE VIEW view_employee  
AS SELECT job_id, emp_no, e_name  
FROM employee  
WHERE emp_no LIKE '%43567';
```



Results	Explain	Describe	Saved SQL	History
view created.				
0.02 seconds				

## VIEW:

SELECT \* FROM employee

ORDER BY e\_name;

Results

Explain

Describe

Saved SQL

History

EMP_NO	E_NAME	E_ADDRESS	E_PHN_NO	DEPT_NO	DEPT_NAME	JOB_ID	SALARY
123	lucky	nellore	-	45	math	1	12345
108	ragni	hyd	987878210	13	social	9	900000
101	raju	nagpur	9876543210	23	maths	1	7890
145	ram	vizag	9678543210	-	telugu	9	45678
105	rani	guduru	9878543210	13	science	5	70890

rows returned in 0.00 seconds

Download

## SIMPLE VIEW:

CREATE OR REPLACE VIEW view\_employee

AS SELECT job\_id, emp\_no, e\_name

FROM employee

WHERE emp\_no LIKE '%43567';

Results	Explain	Describe	Saved SQL	History
View created.				
0.00 seconds				

### Simple view with columns :

```
CREATE OR REPLACE VIEW view_employee
AS SELECT job_id as id , emp_no as e_no, e_name as name
FROM employee
WHERE emp_no LIKE '%43567';
```

Results	Explain	Describe	Saved SQL	History
View created.				
0.02 seconds				

### MODIFYING A VIEW:

```
CREATE OR REPLACE VIEW view_employee
AS SELECT emp_no,job_id,e_name,salary
FROM employee
WHERE emp_no LIKE '%43567';
```

Results

Explain

Describe

Saved SQL

History

View created.

0.02 seconds

## SECTION 16

### CREATING A SEQUENCE

```
CREATE SEQUENCE runner_id_seq  
  
INCREMENT BY 1  
  
START WITH 1  
  
MAXVALUE 50000  
  
NOCACHE  
  
NOCYCLE;
```

Results

Explain

Describe

Saved SQL

History

Sequence created.

0.02 seconds

### Confirming Sequences

```
SELECT sequence_name, min_value, max_value, increment_by,  
last_number  
  
FROM user_sequences;
```

Results

Explain

Describe

Saved SQL

History

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
RUNNER_ID_SEQ	1	50000	1	1

1 rows returned in 0.01 seconds

Download

