**CONSTRAINTS**

**Queries**:

SQL>create table employee (empno number(9), name varchar(15),job varchar(20),sal number(10),commission number(10),department varchar(10), primary Key(empno));

Table created.

SQL>Insert into employee values(1,'Ram','manager',20000,2000,'IT');

1 row created.

SQL>alter table employee add constraint chk\_sal CHECK(sal>=0 and sal>commission);

Table altered.

SQL> select constraint\_name,constraint\_type , table\_name from user\_constraints where table\_name='EMPLOYEE';

CONSTRAINT\_NAME C TABLE\_NAME

------------------------------ - ------------------------------

SYS\_C008776 P EMPLOYEE

CKH\_SAL C EMPLOYEE

**VIEWS**

**Queries:**

SQL> insert into employee values(29, 'nitin','manager',29000,2900,'hr');

1 row created.

SQL> select \* from employee;

EMPNO NAME JOB SALARY COMMISSION DEPARMENT

---------- --------------- --------------- ---------- ---------- ---------------

29 nithes manager 29000 2900 hr

20 siva developer 60000 2000 software

6 harish tester 100000 6000 software

2 tom trainee 30000 3000 software

SQL> create view workers as select \* from employee where job <> 'manager';

View created.

SQL> select \* from workers;

EMPNO NAME JOB SALARY COMMISSION DEPARMENT

---------- --------------- --------------- ---------- ---------- ---------------

20 siva developer 60000 2000 software

6 harish tester 100000 6000 software

2 tom trainee 30000 3000 software

SQL> create view upd\_workers as select \* from employee;

View created.

SQL> insert into upd\_workers values(07,'shriram','test associate',25000,2500,'IT','07-mar-00',22);

1 row created.

**PARTITIONS**

**QUERIES:**

SQL> create table sal\_part(emp\_id number primary key,name varchar(20), Sal number(10,2)) partition by range(Sal)(partition p1 values less than(5000) tablespace users,partition p2 values less than(15000) tablespace users);

Table created.

SQL> Insert into sal\_part values(1000,'Ram',10000);

1 row created.

SQL> Insert into sal\_part values(1001,'lal',3000);

1 row created.

SQL> Select \* from sal\_part partition(p1);

EMP\_ID NAME SAL

---------- -------------------- ----------

1001 lal 3000

SQL> Select \* from sal\_part partition(p2);

EMP\_ID NAME SAL

---------- -------------------- ----------

1000 Ram 10000

SQL> Select tablespace\_name from user\_tables;

TABLESPACE\_NAME

------------------------------

USERS

USERS

USERS

USERS

**INDEX**

**Queries:**

create index index\_name on employee(name);

Index created.

create index index\_upper\_name on employee(LOWER(name));

Index created.

SELECT INDEX\_NAME FROM USER\_INDEXES;

INDEX\_NAME

------------------------------

INDEX\_UPPER\_NAME

IDX\_UPPER\_NAME

ALTER INDEX IDX\_UPPER\_NAME REBUILD;

Index altered.

DROP INDEX INDEX\_NAME;

Index dropped.

**SEQUENCES**

**QUERIES:**

create sequence sequ\_no start with 999 minvalue 1 maxvalue 1000000 nocycle;

Sequence created.

select sequence\_name from user\_sequences;

SEQUENCE\_NAME

------------------------------

SEQU\_NO

insert into employee values(sequ\_no.nextval,'manoj','associate',15000,3500,'IT','02-mar-12',45);

1 row created.

SQL> select sequ\_no.nextval from dual;

NEXTVAL

----------

1019

SQL> select sequ\_no.currval from dual;

CURRVAL

----------

1019

**USERS AND PRIVILEGE**

**Queries**:

create user test01 identified by kiwi;

User created.

grant create session to test01;

Grant succeeded.

GRANT RESOURCE TO test01;

Grant succeeded.

SQL> grant all on employee to test01;

Grant succeeded.

SQL> revoke delete on employee from test01;

Revoke succeeded.

**SOLVING QUERIES**

**QUERIES:**

SQL> create table personal(EmpNo number(10) ,bloodgrp varchar(5),height number(3),weight number(5),mobilenum number (12));

Table created.

SQL>

SQL> alter table personal add foreign key FK\_EMPNO (EmpNo) references employee(EmpNo);

Table altered.

SQL> select constraint\_name,constraint\_type , table\_name from user\_constraints where table\_name='PERSONAL';

CONSTRAINT\_NAME C TABLE\_NAME

------------------------------ - ------------------------------

FK\_EMPNO R PERSONAL

SQL> insert into personal values(2,'o+ve',210,60, 989988989);

1 row created.

SQL> insert into personal values(3,'b+ve',170,60, 68686868);

1 row created.

SQL> select \* from personal;

EMPNO BLOODGRP HEIGHT WEIGHT MOBILENUM

---------- ---------- ---------- ---------- ----------

2 o+ve 210 60 989988989

3 b+ve 170 60 68686868

SQL> select name from employee where age>40 ;

NAME

---------------

manoj

baldan

SQL> select name,sal from employee where sal between 10000 and 60000;

NAME SAL

--------------- ----------

Ram 20000

latha 12000

kumar 10000

manoj 15000

baldan 30000

manoj 15000

select employee.name ,employee.age ,personal.bloodgrp from employee,personal where employee.empno=personal.empno and bloodgrp='o+ve';

NAME AGE BLOODGRP

--------------- ---------- ----------

latha 19 o+ve

SQL> select employee.name ,personal.height from employee,personal where employee.empno=personal.empno and height=(select max(height) from personal);

NAME HEIGHT

--------------- ----------

latha 210

**PROCEDURE**

**QUERIES:**

create procedure pr\_add\_emp(name varchar,job varchar,sal number,commission number , department varchar, dob date,age number)

is

begin

insert into employee values(seq\_no.nextval,name,job,sal,commission,department,dob,age);

end;

/

Procedure created.

SQL> set serverout on

SQL> execute pr\_add\_emp('kumar','analyst','50000','5000','it','05-sep-04',17);

PL/SQL procedure successfully completed.

**DECODE**

**QUERIES:**

SQL> create or replace procedure num2word\_dec (n number) is

word varchar(20);

begin

select decode (n,0,'zero',1,'one',2,'two',3,'three',4,'four',5,'five',6,'six',7,'seven',

8,'eight',9,'nine','non single digit') into word from dual;

dbms\_output.put\_line(word);

end;

/

Procedure created.

SQL> set serverout on

SQL> execute num2word\_dec(2);

two

PL/SQL procedure successfully completed.

**PACKAGE**

**QUERIES:**

SQL> create package pack1 is

function fn\_add2(first number , second number)

return number;

procedure pr\_year(db date);

end;

/

Package created.

SQL> create package body pack1 is

function fn\_add2(first number,second number)

return number is

begin

return(first+second);

end;

procedure pr\_year(db date) is

begin

DBMS\_output.put\_line((sysdate-db)/365);

end;

end;

/

Package body created.

SQL> select pack1.fn\_add2(10,20) from dual;

PACK1.FN\_ADD2(10,20)

--------------------

30

SQL> execute pack1.pr\_year('20-nov-2000');

PL/SQL procedure successfully completed.

**TRIGGER**

**QUERIES:**

SQL> create table emp\_log as select \* from employee where empno=0;

Table created.

SQL> alter table emp\_log add(new\_dob date,user\_id varchar(10),curdate date);

Table altered.

SQL> create or replace trigger trig\_update after update on employee for each row

begin

insert into emp\_log values(:old.empno,:old.name,:old.job,:old.salary,:old.commission,

:old.deparment,:old.dob,:old.age,:new.dob,user,sysdate);

end;

/

Trigger created.

SQL> update employee set dob='16-jan-2003' where empno=1;

1 row updated.

**TABLESPACE**

**QUERIES:**

SQL> create tablespace tbs\_tom

2 datafile 'tbs\_tom\_01.dat'

3 SIZE 20M

4 ONLINE;

Tablespace created.

SQL> create table tbs\_table(deptno varchar(10),name varchar(10),age number(4)) tablespace tbs\_tom;

Table created.

SQL> insert into tbs\_table values('19-uca-051','tom',21);

1 row created.

SQL> alter tablespace tbs\_tom

2 add datafile 'tbs\_tom\_02.dat'

3 SIZE 20M

4 AUTOEXTEND ON;

Tablespace altered.

SQL> alter tablespace tbs\_tom

2 drop datafile 'tbs\_srini\_02.dat';

Tablespace altered.

SQL> select tablespace\_name,table\_name from user\_tables where table\_name='TBS\_TABLE';

TABLESPACE\_NAME TABLE\_NAME

------------------------------ ------------------------------

TBS\_TOM TBS\_TABLE

TABLESPACE\_NAME

------------------------------

TBS\_TOM

**CLUSTERS**

**QUERIES:**

SQL> create cluster tom\_clust (empno number(10)) size 600 tablespace tbs\_tom;

Cluster created.

SQL> CREATE INDEX idx\_tom\_clust ON CLUSTER tom\_clust;

Index created.

SQL> CREATE TABLE emp\_10

2 CLUSTER tom\_clust (empno)

3 AS SELECT \* FROM employee WHERE empno = 10;

Table created.

SQL> CREATE TABLE emp\_20

2 CLUSTER tom\_clust (empno)

3 AS SELECT \* FROM employee WHERE empno = 20;

Table created.

SQL> select cluster\_name , tablespace\_name from user\_clusters;

CLUSTER\_NAME TABLESPACE\_NAME

------------------------------ ------------------------------

TOM\_CLUST TBS\_TOM

**DEADLOCK**

**QUERIES:**

SQL> grant all on personal to cat04;

Grant succeeded.

SQL> lock table personal in share mode ;

Table(s) Locked.

SQL> select \* from personal;

EMPNO BLOODGRP HEIGHT WEIGHT MOBILENUM

---------- ---------- ---------- ---------- ----------

2 o+ve 150 60 989988989

3 b+ve 200 60 68686868

1 B+ve 270 70 9896768989

13 rows selected.

update personal set height=200 where empno=3;

ERROR at line 1:

ORA-00060: deadlock detected while waiting for resource

SQL> commit;

Commit complete.

**QUERY TUNING**

**QUERIES:**

13:52:31 SQL> set timing on

Elapsed: 00:00:00.01

13:55:14 SQL> select count(\*) rows\_query1 from emp where sal>10000;

ROWS\_QUERY1

-----------

5

Elapsed: 00:00:00.04

13:56:41 SQL> select count(\*) rows\_query2 from emp where job='Manager'

13:56:55 2 ;

ROWS\_QUERY2

-----------

2

**BACKUP**

**QUERIES:**

Microsoft Windows [Version 10.0.19044.1706]

(c) Microsoft Corporation. All rights reserved.

C:\>exp

Export: Release 10.2.0.3.0 - Production on Thu May 19 16:01:12 2022

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Username: cat01@orcl

Password:

Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production

With the Partitioning, OLAP and Data Mining options

Enter array fetch buffer size: 4096 > 4096

Export file: EXPDAT.DMP > d:\exp

(2)U(sers), or (3)T(ables): (2)U > u

Export grants (yes/no): yes > yes

Export table data (yes/no): yes > yes

Compress extents (yes/no): yes > yes

Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set

. exporting pre-schema procedural objects and actions

. exporting foreign function library names for user CAT01

. exporting PUBLIC type synonyms

. exporting private type synonyms

. exporting object type definitions for user CAT01

About to export CAT01's objects ...

. exporting database links

. exporting sequence numbers

. exporting cluster definitions

. about to export CAT01's tables via Conventional Path ...

. . exporting table ACC 0 rows exported

. . exporting table E1 4 rows exported

. . exporting table EMP 6 rows exported

. . exporting table EMP0 3 rows exported

. . exporting table EMP1 5 rows exported

. . exporting table EMPART

. . exporting partition P1 0 rows exported

. . exporting partition P2 0 rows exported

. . exporting table EMPART1

. . exporting partition P1 0 rows exported

. . exporting partition P2 0 rows exported

. . exporting table EMPLOY 1

rows exported

. . exporting table EMPY 1 rows exported

. . exporting table EMP\_TB 0 rows exported

. . exporting table PERSONNEL 5 rows exported

. exporting synonyms

. exporting views

. exporting stored procedures

. exporting operators

. exporting referential integrity constraints

. exporting triggers

. exporting indextypes

. exporting bitmap, functional and extensible indexes

. exporting posttables actions

. exporting materialized views

. exporting snapshot logs

. exporting job queues

. exporting refresh groups and children

. exporting dimensions

. exporting post-schema procedural objects and actions

. exporting statistics

Export terminated successfully without warnings.

C:\>