**Create/Drop/Describe Table**

create table department (dept\_id number(7) primary key, dept\_name varchar2(50) unique, regi\_date date default sysdate);

create table contact (con\_id number(7) primary key, mobile number(15) unique, country varchar2(50) default 'Bangladesh')

create table employee (id number(7) primary key, emp\_name varchar2(20) not null, salary number(7,2) check(salary>=25000), joining\_date date check(joining\_date<='01-Jan-2000'), dept\_id number(7) not null, con\_id number(7), constraint emp\_dep foreign key (dept\_id) references department, constraint emp\_con foreign key (con\_id) references contact);

create table department\_backup as select \* from department;

drop table dhaka;

desc dhaka;

**Create Role/User**

create role super\_user;

create user shuvo identified by shuvo;

**Grant Privilege**

grant create session, create table, create view, create sequence, create procedure, create synonym, create trigger to super\_user;

grant super\_user to shuvo;

grant dba to shuvo;

**Grant Tablespace to User**

grant unlimited tablespace to shuvo;

alter user olive quota 500m on system;

**Insert Data**

insert into department values (dept\_seq.nextval, 'Administration');

insert into contact (con\_id, mobile, country) values(con\_seq.nextval, 01712345678, 'Japan');

insert into employee (id, emp\_name, salary, joining\_date, dept\_id, con\_id) values(emp\_seq.nextval, 'AAA', 25000, '01-Feb-1998', 10, 10);

**Table Privilege/Add Constraints**

ALTER TABLE dhaka READ ONLY;

ALTER TABLE dhaka READ WRITE;

ALTER TABLE department ADD (job\_id VARCHAR2(9));

ALTER TABLE department DROP COLUMN job\_id;

alter table department modify (dept\_name varchar2(55));

alter table employee add constraints emp\_dept\_id unique (con\_id);

**Update Data**

update dhaka set name='C' where income = 30000;

update employee set salary=35000, emp\_name='BBB' where con\_id=15;

update company set brand\_id = (select brand\_id from company where name='Google Inc.') where id=03;

**Create View/Sequence/Synonym/Index**

create sequence dept\_seq increment by 10 start with 10 maxvalue 5000 nocache nocycle;

create sequence con\_seq increment by 5 start with 5 nocache nocycle;

create sequence emp\_seq increment by 1 start with 1 nocache nocycle;

create view dha\_view1 as select id, name, email from dhaka;

create or replace view empvu80 (id\_number, name, sal, department\_id) as select employee\_id, first name|| ' ' || last\_name, salary, department\_id from employees;

CREATE OR REPLACE VIEW dept\_sum\_vu (name, minsal, maxsal, avgsal) AS SELECT d.department\_name, MIN(e.salary), MAX(e.salary), AVG(e.salary) FROM employees e JOIN departments d ON (e.department\_id = d.department\_id) GROUP BY d.department\_name;

CREATE INDEX emp\_last\_name\_idx ON employees(last\_name);

create synonym e20 for empvu80;

**Show View/Table/Sequence/Username**

select view\_name from user\_views;

select table\_name from user\_tables;

select sequence\_name from user\_sequences;

select USERNAME from DBA\_USERS;

**Autocommit**

show autocommit;

set autocommit on;

**Delete Data**

delete brand where id=04;

delete from department;

truncate table brand; (Works with DDL Statement)

**Spool On/Off**

spool E:/quiz.txt;

spool off;

**Create Trigger for Insert**

**Step-1: Create Two Tables**

create table usa (id number (5), fname varchar2(55), lname varchar2(55));

create table usa\_backup as select \* from usa;

**Step-2: Create Trigger:**

CREATE OR REPLACE TRIGGER usa\_trigger AFTER insert ON usa (**NB: after update/before delete**)

FOR EACH ROW

DECLARE

BEGIN

insert into usa\_backup values (:new.id, :new.fname, :new.lname);

dbms\_output.put\_line ('Data inserted successfully on usa\_backup table');

END;

/

**Step-3: Insert Data & View from Backup Table**

insert into usa values (01,'Mehedi','Hasan');

select \* from usa\_backup;

**Create Trigger for Update**

**Step-1: Create Trigger**

CREATE OR REPLACE TRIGGER usa\_trigger\_update AFTER update ON usa

FOR EACH ROW

DECLARE

BEGIN

UPDATE usa\_backup

set id = :new.id, name = :new.name, lname = :new.lname

where id = :old.id or name = :old.name or lname = :old.lname;

dbms\_output.put\_line ('Data successfully updated into usa\_backup table');

END;

**Step-2: Update Main Table & View from Backup Table**

update usa set name = 'Hollywood' where id = 5010;

update usa set id = 5010 where name = 'Hollywood';

select \* from usa\_backup;

**Create Trigger for Delete**

**Step-1: Create Trigger**

CREATE OR REPLACE TRIGGER usa\_trigger\_delete AFTER delete ON usa

FOR EACH ROW

DECLARE

BEGIN

DELETE from usa\_backup

where id = :old.id or name = :old.name;

dbms\_output.put\_line ('Data successfully deleted from usa\_backup table');

END;

**Step-2: Delete from Main Table & View from Backup Table**

delete from usa where name = 'Los Angeles';

delete from usa where id = 5010;

select \* from usa\_backup;

**Create Procedure for Insert**

**Step-1: Create Table**

Create Table Customer (id number(5) primary key, name varchar2(55) not null, mobile varchar2(55), country varchar2(55));

**Step-2: Create Insert Procedure**

CREATE OR REPLACE PROCEDURE insertCustomer(

p\_id IN customer.id%TYPE,

p\_name IN customer.name%TYPE,

p\_mobile IN customer.mobile%TYPE,

p\_country IN customer.country%TYPE)

IS

BEGIN

INSERT into customer (id, name, mobile, country)

VALUES (p\_id, p\_name, p\_mobile, p\_country);

COMMIT;

END;

/

**Step-3: Calling Insert Procedure**

BEGIN

insertCustomer(101, 'Shuvo', 01711000001, 'Bangladesh');

END;

/

**Create Procedure for Update**

**Step-1: Create Update Procedure**

CREATE OR REPLACE PROCEDURE updateCustomer(

p\_id IN customer.id%TYPE,

p\_name IN customer.name%TYPE,

p\_mobile IN customer.mobile%TYPE,

p\_country IN customer.country%TYPE)

IS

BEGIN

UPDATE customer SET name = p\_name, mobile = P\_mobile, country = p\_country where id = p\_id or name = p\_name;

COMMIT;

END;

/

**Step-2: Calling Update Procedure**

BEGIN

updateCustomer(102,'Masud',01711000002,'France');

END;

/

**Create procedure for delete**

**Step-1: Create Update Procedure**

CREATE OR REPLACE PROCEDURE deleteCustomer(

p\_id IN customer.id%TYPE,

p\_name IN customer.name%TYPE)

IS

BEGIN

DELETE from customer where id = p\_id or name = p\_name;

COMMIT;

END;

/

**Step-2: Calling Delete Procedure**

BEGIN

deleteCustomer(103, 'Rafin');

END;

/