Activities:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| activity\_id | NUMBER(2) | pk\_activites |  |
| name | VARCHAR2(20) | UPPER |  |
| start\_date | DATE |  | sysdate |

Stages:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| stage\_id | NUMBER(2) | pk\_stages |  |
| stage\_type | VARCHAR2(15) | NOT NULL |  |
| location | VARCHAR2(15) |  | ‘UK’ |

Plans:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| \*activity\_id | NUMBER(2) | FK(fk\_p\_activities), PK(pk\_plans) |  |
| \*stage\_id | NUMBER(2) | FK(fk\_p\_stages), PK(pk\_plans) |  |
| plan\_id | NUMBER(4) | pk\_plans |  |
| start\_date | DATE | NOT NULL |  |
| end\_date | DATE | NOT NULL |  |

Resources:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| resource\_id | NUMBER(2) | pk\_resources |  |
| resource\_name | VARCHAR2(8) | UPPER |  |
| resource\_type | VARCHAR2(10) | UNIQUE |  |
| description | VARCHAR2(30) | NOT NULL |  |
| item\_cost | NUMBER(6) |  |  |

Activity\_resources:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| \*activity\_id | NUMBER(2) | fk\_a\_r\_activities(optional) |  |
| \*resource\_id | NUMBER(2) | FK(fk\_a\_r\_resources), PK(pk\_activity\_resources) |  |
| item\_no | NUMBER(3) | PK(pk\_activity\_resources) |  |
| item\_required | VARCHAR2(10) | NOT NULL |  |
| item\_quantity | NUMBER(6) | NOT NULL |  |

Costs:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| \*activity\_id | NUMBER(2) | fk\_c\_activities, NOT NULL |  |
| cost\_id | NUMBER(2) | pk\_costs |  |
| item\_description | VARCHAR2(15) |  |  |
| amount | NUMBER(8) |  |  |

Staffs:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| staff\_id | NUMBER(2) | PK(pk\_staffs) |  |
| first\_name | VARCHAR2(10) | UPPER |  |
| last\_name | VARCHAR2(10) | UPPER |  |
| contact\_number | VARCHAR2(11) | UNIQUE |  |
| email | VARCHAR2(35) | UNIQUE |  |
| town | VARCHAR2(15) |  | BEDFORD |
| country | VARCHAR2(15) |  | ‘UK’ |
| date\_registered | DATE |  | sysdate |
| gender | CHAR(1) | CHECK IN (‘M’ OR ‘F’) |  |

Roles:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| role\_id | NUMBER(2) | pk\_roles |  |
| role\_type | VARCHAR2(12) | UNIQUE |  |
| description | VARCHAR2(30) | NOT NULL |  |

Staff\_roles:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| \*staff\_id | NUMBER(2) | FK(fk\_s\_r\_staffs), PK(pk\_staff\_roles) |  |
| \*role\_id | NUMBER(2) | FK(fk\_s\_r\_roles),  PK(pk\_staff\_roles) |  |
| staff\_role\_id | NUMBER(4) | pk\_staff\_roles |  |
| started\_date | DATE | NOT NULL |  |
| duty\_status | VARCHAR2(12) | CHECK IN ( active) |  |

Groups:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| \*activity\_id | NUMBER(2) | FK(fk\_g\_activities),  PK(pk\_groups) |  |
| \*staff\_role\_id | NUMBER(4) | FK(fk\_g\_staff\_roles)  PK(pk\_groups) |  |
| group\_id | NUMBER(6) | pk\_groups |  |
| \*staff\_id | NUMBER(2) | FK(fk\_g\_staff\_roles) |  |
| \*role\_id | NUMBER(2) | FK(fk\_g\_staff\_roles) |  |
| status | VARCHAR2(10) | CHECK IN (‘active’) |  |
| group\_size | NUMBER(3) | NOT NULL |  |

Reviews:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| \*plan\_id | NUMBER(4) | fk\_r\_plans (optional) |  |
| \*stage\_id | NUMBER(2) | fk\_r\_plans(optional) |  |
| \*activity\_id | NUMBER(2) | fk\_r\_groups(optional) |  |
| \*staff\_role\_id | NUMBER(4) | fk\_r\_groups(optional) |  |
| \*group\_id | NUMBER(6) | fk\_r\_groups (optional) |  |
| review\_id | NUMBER(2) | pk\_reviews |  |
| time | TIMESTAMP | NOT NULL |  |
| location | VARCHAR2(12) | NOT NULL |  |

/\* TIMESTAMP stores in format (YYYY-MM-DD HH24:MI:SS.FF)

* (.FF) is optional

Example:

(1995-01-24 09:26:50)

\*/

Actions:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| \*review\_id | NUMBER(2) | fk\_a\_reviews (OPTIONAL) |  |
| action\_id | NUMBER(2) | pk\_actions |  |
| owner\_name | VARCHAR2(12) | UPPER |  |
| deadline | DATE | NOT NULL |  |

/\* (Group\_13), 2021 CSY1026 \*/

/\*----------CREATING TABLE ACTIVITIES------------\*/

CREATE TABLE Activities(

activity\_id NUMBER(2),

name VARCHAR2(20), start\_date DATE);

/\*----------CREATING TABLE Stages------------\*/

CREATE TABLE Stages(

stage\_id NUMBER(2),

stage\_type VARCHAR2(15) NOT NULL,

location VARCHAR2 (15));

/\*----------CREATING TABLE PLANS------------\*/

CREATE TABLE plans(

activity\_id NUMBER(2), stage\_id NUMBER(2), plan\_id NUMBER(4), start\_date DATE NOT NULL, end\_date DATE NOT NULL);

/\* PRIMARY KEY and FOREIGN KEY can be defined at the time of creating table

but it is not good practice.

CREATE TABLE PLANS(

activity\_id NUMBER(2),

stage\_id NUMBER(2),

plan\_id NUMBER(4),

start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

CONSTRAINT pk\_plans PRIMARY KEY (plan\_id),

CONSTRAINT fk\_p\_activities

FOREIGN KEY (activity\_id)

REFERENCES activities(activity\_id),

CONSTRAINT fk\_p\_stages

FOREIGN KEY (stage\_id)

REFERENCES stages(stage\_id));

\*/

/\*----------CREATING TABLE RESOURCES------------\*/

CREATE TABLE RESOURCES(

resource\_id NUMBER(2),

resource\_name VARCHAR2(8),

resource\_type VARCHAR2(10),

description VARCHAR2(30) NOT NULL,

item\_cost NUMBER(6));

/\*----------CREATING TABLE ACTIVITY\_RESOURCES------------\*/

CREATE TABLE ACTIVITY\_RESOURCES(activity\_id NUMBER(2), resource\_id NUMBER(2), item\_no NUMBER(3),

item\_required VARCHAR2(10) NOT NULL,

item\_quantity NUMBER(6) NOT NULL);

/\*----------CREATING TABLE COSTS------------\*/

CREATE TABLE COSTS(

activity\_id NUMBER(2) NOT NULL,

cost\_id NUMBER(2),

item\_description VARCHAR2(15),

amount NUMBER(8));

/\*----------CREATING TABLE STAFFS------------\*/

CREATE TABLE STAFFS(

staff\_id NUMBER(2),

first\_name VARCHAR2(10),

last\_name VARCHAR2(10),

contact\_number NUMBER(11) NOT NULL,

email VARCHAR2(35),

town VARCHAR2(15),

country VARCHAR(15),

date\_registered DATE,

gender CHAR(1));

/\*----------CREATING TABLE ROLES------------\*/

CREATE TABLE ROLES(

role\_id NUMBER(2),

role\_type VARCHAR2(12),

description VARCHAR2(30) NOT NULL);

/\*----------CREATING TABLE STAFF\_ROLES------------\*/

CREATE TABLE STAFF\_ROLES(

staff\_id NUMBER(2), role\_id NUMBER(2),

staff\_role\_id NUMBER(4),

started\_date DATE NOT NULL,

duty\_status VARCHAR2(12));

/\*----------CREATING TABLE GROUPS------------\*/

CREATE TABLE groups(

activity\_id NUMBER(2), staff\_role\_id NUMBER(4), group\_id NUMBER(6),

staff\_id NUMBER(2), role\_id NUMBER(2),

status VARCHAR2(10), group\_size NUMBER(3) NOT NULL );

/\*----------CREATING TABLE REVIEWS------------\*/

CREATE TABLE REVIEWS(

plan\_id NUMBER(4), stage\_id NUMBER(2), activity\_id NUMBER(2), staff\_role\_id NUMBER(4), group\_id NUMBER(6),

review\_id NUMBER(2),

time TIMESTAMP NOT NULL,

location VARCHAR2(12) NOT NULL );

/\*----------CREATING TABLE ACTIONS------------\*/

CREATE TABLE ACTIONS(

review\_id NUMBER(2), action\_id NUMBER(2),

owner\_name VARCHAR2(12),

deadline DATE NOT NULL);

/\*----------CREATING SEQUQNCE TABLE ACTIVITIES------------\*/

CREATE SEQUENCE Activity\_sequence

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 10

CACHE 10;

/\*----------CREATING SEQUQNCE TABLE STAGE------------\*/

CREATE SEQUENCE Stage\_sequence

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 10

CACHE 10;

/\*----------CREATING SEQUQNCE TABLE PLANS------------\*/

CREATE SEQUENCE Plan\_sequence

START WITH 11

INCREMENT BY 1

MINVALUE 11

MAXVALUE 1010

CACHE 10;

/\*----------CREATING SEQUQNCE TABLE RESOURCES------------\*/

CREATE SEQUENCE Resource\_sequence

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 10

CACHE 10;

/\*---ACTIVITY\_RESOURCES sequence ---\*/

CREATE SEQUENCE a\_r\_sequence

START WITH 11

INCREMENT BY 1

MINVALUE 11

MAXVALUE 101

CACHE 10;

/\*----------CREATING SEQUQNCE TABLE COSTS------------\*/

CREATE SEQUENCE Cost\_sequence

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 10

CACHE 10;

/\*----------CREATING SEQUQNCE TABLE STAFFS------------\*/

CREATE SEQUENCE STAFF\_sequence

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 10

CACHE 10;

/\*----------CREATING SEQUENCE TABLE ROLES------------\*/

CREATE SEQUENCE Role\_sequence

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 10

CACHE 10;

/\*---staff\_roles sequence ---\*/

CREATE SEQUENCE s\_r\_sequence

START WITH 11

INCREMENT BY 1

MINVALUE 11

MAXVALUE 1010

CACHE 10;

/\*----------CREATING SEQUENCE TABLE GROUPS------------\*/

CREATE SEQUENCE Group\_sequence

START WITH 111

INCREMENT BY 1

MINVALUE 111

MAXVALUE 101010

CACHE 10;

/\*----------CREATING SEQUENCE TABLE REVIEWS------------\*/

CREATE SEQUENCE review\_sequence

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 10

CACHE 10;

/\*-----action sequence --- \*/

CREATE SEQUENCE action\_sequence

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 10

CACHE 10;

/\* (Group\_13), 2021 CSY1026 \*/

/\* to add primary key constraints \*/

ALTER TABLE activities

ADD CONSTRAINT pk\_activities

PRIMARY KEY (activity\_id);

ALTER TABLE stages

ADD CONSTRAINT pk\_stages

PRIMARY KEY (stage\_id);

ALTER TABLE plans

ADD CONSTRAINT fk\_p\_activities

FOREIGN KEY (activity\_id)

REFERENCES activities (activity\_id);

ALTER TABLE plans

ADD CONSTRAINT fk\_p\_stages

FOREIGN KEY (stage\_id)

REFERENCES stages (stage\_id);

ALTER TABLE plans

ADD CONSTRAINT pk\_plans

PRIMARY KEY (activity\_id, stage\_id, plan\_id);

ALTER TABLE resources

ADD CONSTRAINT pk\_resources

PRIMARY KEY (resource\_id);

ALTER TABLE activity\_resources

ADD CONSTRAINT fk\_a\_r\_resources

FOREIGN KEY (resource\_id)

REFERENCES resources (resource\_id);

ALTER TABLE activity\_resources

ADD CONSTRAINT pk\_activity\_resources

PRIMARY KEY (resource\_id, item\_no);

ALTER TABLE activity\_resources

ADD CONSTRAINT fk\_a\_r\_activities

FOREIGN KEY (activity\_id)

REFERENCES activities (activity\_id);

ALTER TABLE costs

ADD CONSTRAINT pk\_costs

PRIMARY KEY (cost\_id);

ALTER TABLE costs

ADD CONSTRAINT fk\_c\_activities

FOREIGN KEY (activity\_id)

REFERENCES activities (activity\_id);

ALTER TABLE staffs

ADD CONSTRAINT pk\_staffs

PRIMARY KEY (staff\_id);

ALTER TABLE roles

ADD CONSTRAINT pk\_roles

PRIMARY KEY (role\_id);

ALTER TABLE staff\_roles

ADD CONSTRAINT fk\_s\_r\_staffs

FOREIGN KEY (staff\_id)

REFERENCES staffs (staff\_id);

ALTER TABLE staff\_roles

ADD CONSTRAINT fk\_s\_r\_roles

FOREIGN KEY (role\_id)

REFERENCES roles (role\_id);

ALTER TABLE staff\_roles

ADD CONSTRAINT pk\_staff\_roles

PRIMARY KEY (staff\_id, role\_id, staff\_role\_id);

ALTER TABLE groups

ADD CONSTRAINT fk\_g\_activities

FOREIGN KEY (activity\_id)

REFERENCES activities (activity\_id);

ALTER TABLE groups

ADD CONSTRAINT fk\_g\_staff\_roles

FOREIGN KEY (staff\_role\_id, staff\_id, role\_id)

REFERENCES staff\_roles (staff\_role\_id, staff\_id, role\_id);

ALTER TABLE groups

ADD CONSTRAINT pk\_groups

PRIMARY KEY (activity\_id, staff\_role\_id, group\_id);

ALTER TABLE reviews

ADD CONSTRAINT pk\_reviews

PRIMARY KEY (review\_id);

ALTER TABLE reviews

ADD CONSTRAINT fk\_r\_plans

FOREIGN KEY (plan\_id, stage\_id, activity\_id)

REFERENCES plans (plan\_id, stage\_id, activity\_id);

ALTER TABLE reviews

ADD CONSTRAINT fk\_r\_groups

FOREIGN KEY (activity\_id, staff\_role\_id, group\_id)

REFERENCES groups (activity\_id, staff\_role\_id, group\_id);

ALTER TABLE actions

ADD CONSTRAINT fk\_a\_reviews

FOREIGN KEY (review\_id)

REFERENCES reviews (review\_id);

ALTER TABLE actions

ADD CONSTRAINT pk\_actions

PRIMARY KEY (action\_id);

/\* To add CHECK constraints \*/

ALTER TABLE activities

ADD CONSTRAINT ck\_activities\_name\_upper

CHECK (name = UPPER(name));

ALTER TABLE resources

ADD CONSTRAINT ck\_resources\_name\_upper

CHECK (resource\_name = UPPER(resource\_name));

ALTER TABLE staffs

ADD CONSTRAINT ck\_staffs\_first\_name\_upper

CHECK (first\_name = UPPER(first\_name));

ALTER TABLE staffs

ADD CONSTRAINT ck\_staffs\_last\_name\_upper

CHECK (last\_name = UPPER(last\_name));

ALTER TABLE staffs

ADD CONSTRAINT ck\_staffs\_gender

CHECK (gender = 'M' OR gender = 'F');

ALTER TABLE staff\_roles

ADD CONSTRAINT ck\_staff\_roles\_duty\_status

CHECK (duty\_status = 'Active');

ALTER TABLE groups

ADD CONSTRAINT ck\_groups\_status

CHECK (status = 'Active');

ALTER TABLE actions

ADD CONSTRAINT ck\_actions\_owner\_name\_upper

CHECK (owner\_name = UPPER(owner\_name));

/\*------To add UNIQUE CONSTRAINTS------ \*/

ALTER TABLE resources

ADD CONSTRAINT u\_resources

UNIQUE (resource\_type);

ALTER TABLE staffs

ADD CONSTRAINT u\_staffs

UNIQUE (contact\_number, email);

ALTER TABLE roles

ADD CONSTRAINT u\_roles

UNIQUE (role\_type);

/\* to add default value \*/

ALTER TABLE activities

MODIFY start\_date DEFAULT SYSDATE;

ALTER TABLE stages

MODIFY location DEFAULT 'UK';

ALTER TABLE staffs

MODIFY town DEFAULT 'BEDFORD';

ALTER TABLE staffs

MODIFY country DEFAULT 'UK';

ALTER TABLE staffs

MODIFY date\_registered DEFAULT SYSDATE;

/\* (Group\_13), 2021 CSY1026 \*/

INSERT INTO Activities (activity\_id, name, start\_date) VALUES (activity\_sequence.nextval, 'GAME DEVELOPMENT', '07AUG19');

INSERT INTO STAGES (stage\_id, stage\_type, location) VALUES (stage\_sequence.nextval, 'development', 'NORTHAMPTON');

INSERT INTO PLANS (activity\_id, stage\_id, plan\_id, start\_date, end\_date) VALUES (activity\_sequence.currval, stage\_sequence.currval, plan\_sequence.nextval , '05JAN19', '05JAN20');

INSERT INTO STAFFS (staff\_id, first\_name, last\_name, contact\_number, email, town, country, date\_registered, gender) VALUES (staff\_sequence.nextval, 'AYUSH', 'SHRESTHA', '9843243245', 'ayus@gmail.com', 'NORTHAMPTON', 'UK', '01JAN17', 'M');

INSERT INTO ROLES (role\_id, role\_type, description) VALUES (role\_sequence.nextval, 'Management', 'You have to manage projects');

INSERT INTO Staff\_roles (staff\_id, role\_id, staff\_role\_id, started\_date, duty\_status) VALUES (staff\_sequence.currval, role\_sequence.currval, s\_r\_sequence.nextval, '05JAN19', 'Active');

INSERT INTO Groups (activity\_id, staff\_role\_id, group\_id, staff\_id, role\_id, status, group\_size) VALUES (activity\_sequence.currval, s\_r\_sequence.currval, group\_sequence.nextval, staff\_sequence.currval, role\_sequence.currval, 'Active', 10);

INSERT INTO RESOURCES (resource\_id, resource\_name, resource\_type, description, item\_cost) VALUES (resource\_sequence.nextval, 'IR12', 'software', 'IR12for working of software', '123456');

INSERT INTO Activity\_resources (activity\_id, resource\_id, item\_no, item\_required, item\_quantity) VALUES (activity\_sequence.currval , resource\_sequence.currval, 10, 'SOFTWARE', '100' );

INSERT INTO COSTS (activity\_id, cost\_id, item\_description, amount) VALUES (activity\_sequence.currval, cost\_sequence.nextval, 'Power supply', '100');

INSERT INTO Reviews (plan\_id, stage\_id, activity\_id, staff\_role\_id, group\_id, review\_id, time, location) VALUES (plan\_sequence.currval, stage\_sequence.currval, activity\_sequence.currval, s\_r\_sequence.currval, group\_sequence.currval, review\_sequence.nextval, '07AUG19:12:22:58', 'NORTHAMPTON');

INSERT INTO Actions (review\_id, action\_id, owner\_name, deadline) VALUES (review\_sequence.currval, action\_sequence.nextval, 'AMIR', '12OCT18');

INSERT INTO Activities (activity\_id, name, start\_date) VALUES (activity\_sequence.nextval, 'WEB DESIGN', '05JUL21' );

INSERT INTO STAGES(stage\_id, stage\_type, location) VALUES (stage\_sequence.nextval, 'testing', 'NEPAL');

INSERT INTO PLANS (activity\_id, stage\_id, plan\_id, start\_date, end\_date) VALUES (activity\_sequence.currval, stage\_sequence.currval, plan\_sequence.nextval, '12JUL20', '07MAR21');

INSERT INTO STAFFS (staff\_id, first\_name, last\_name, contact\_number, email, town, country, date\_registered, gender) VALUES (staff\_sequence.nextval, 'SIDDHARTH', 'TAMANG', '9832432427', 'sidd@gmail.com', 'BEDFORD', 'UK', '01FEB21', 'M');

INSERT INTO ROLES (role\_id, role\_type, description) VALUES (role\_sequence.nextval, 'Editor', 'For fixing bug of program' );

INSERT INTO Staff\_roles (staff\_id, role\_id, staff\_role\_id, started\_date, duty\_status) VALUES (staff\_sequence.currval, role\_sequence.currval, s\_r\_sequence.nextval, '12APR21', 'Active');

INSERT INTO Groups (activity\_id, staff\_role\_id, group\_id, staff\_id, role\_id, status, group\_size) VALUES (activity\_sequence.currval, s\_r\_sequence.currval, group\_sequence.nextval, staff\_sequence.currval, role\_sequence.currval, 'Active', 2);

INSERT INTO RESOURCES (resource\_id, resource\_name, resource\_type, description, item\_cost) VALUES (resource\_sequence.nextval, 'GO', 'training', 'To train newcommers', '34456');

INSERT INTO Activity\_resources (resource\_id, item\_no, item\_required, item\_quantity) VALUES (resource\_sequence.currval, 14, 'MIC', '2000');

INSERT INTO COSTS (activity\_id, cost\_id, item\_description, amount) VALUES (activity\_sequence.currval, cost\_sequence.nextval, 'MIC for demo', '2000');

INSERT INTO Reviews (review\_id, time, location) VALUES ( review\_sequence.nextval,'15JAN21:06:22:22','UK');

INSERT INTO Actions (review\_id, action\_id, owner\_name, deadline) VALUES (review\_sequence.currval, action\_sequence.nextval,'AMIR', '20DEC21');

INSERT INTO Activities (activity\_id, name, start\_date) VALUES (activity\_sequence.nextval, 'PROGRAM TESTING', '19JAN20');

INSERT INTO STAGES (stage\_id, stage\_type, location) VALUES (stage\_sequence.nextval, 'designing', 'NORTHAMPTON');

INSERT INTO PLANS (activity\_id, stage\_id, plan\_id, start\_date, end\_date) VALUES (activity\_sequence.currval, stage\_sequence.currval, plan\_sequence.nextval, '05AUG19', '07AUG19');

INSERT INTO STAFFS (staff\_id, first\_name, last\_name, contact\_number, email, town, country, date\_registered, gender) VALUES (staff\_sequence.nextval, 'SANDEEPA', 'ACHARYA', '9865656544', 'sandeepa@gmail.com', 'SAKHU', 'NEPAL', '19MAR19', 'F');

INSERT INTO ROLES (role\_id, role\_type, description) VALUES (role\_sequence.nextval, 'Director', 'To direct project in best way');

INSERT INTO Staff\_roles (staff\_id, role\_id, staff\_role\_id, started\_date, duty\_status) VALUES (staff\_sequence.currval, role\_sequence.currval, s\_r\_sequence.nextval, '05AUG21', 'Active');

INSERT INTO Groups (activity\_id, staff\_role\_id, group\_id, status, group\_size) VALUES (activity\_sequence.currval, s\_r\_sequence.currval, group\_sequence.nextval, 'Active', 18);

INSERT INTO RESOURCES (resource\_id, resource\_name, resource\_type, description, item\_cost) VALUES (resource\_sequence.nextval, 'PC', 'management','PC is computer system', '12356');

INSERT INTO Activity\_resources (resource\_id, item\_no, item\_required, item\_quantity) VALUES (resource\_sequence.currval, 12, 'PLATNUM', '2500');

INSERT INTO COSTS (activity\_id, cost\_id, item\_description, amount) VALUES (activity\_sequence.currval, cost\_sequence.nextval, 'Camera', '2500');

INSERT INTO Reviews (review\_id, time, location) VALUES ( review\_sequence.nextval,'28DEC20:12:10:15', 'NEPAL');

INSERT INTO Actions (review\_id, action\_id, owner\_name, deadline) VALUES (review\_sequence.currval, action\_sequence.nextval,'SIDDHARTH', '10JAN19');

INSERT INTO Activities (activity\_id, name, start\_date) VALUES (activity\_sequence.nextval, 'GAME TESTING', '05MAR20');

INSERT INTO STAGES (stage\_id, stage\_type, location) VALUES (stage\_sequence.nextval, 'testing', 'NORTHAMPTON');

INSERT INTO PLANS (activity\_id, stage\_id, plan\_id, start\_date, end\_date) VALUES (activity\_sequence.currval, stage\_sequence.currval, plan\_sequence.nextval, '14MAR17', '14OCT22');

INSERT INTO STAFFS (staff\_id, first\_name, last\_name, contact\_number, email, town, country, date\_registered, gender) VALUES (staff\_sequence.nextval, 'RABIN', 'JOSHI', '9875465232', 'rabin@gmail.com', 'MULPANI', 'USA', '21JUL21', 'M');

INSERT INTO ROLES (role\_id, role\_type, description) VALUES (role\_sequence.nextval, 'Host', 'To host some programs');

INSERT INTO Staff\_roles (staff\_id, role\_id, staff\_role\_id, started\_date, duty\_status) VALUES (staff\_sequence.currval, role\_sequence.currval, s\_r\_sequence.nextval, '19NOV20', 'Active');

INSERT INTO Groups (activity\_id, staff\_role\_id, group\_id, status, group\_size) VALUES (activity\_sequence.currval, s\_r\_sequence.currval, group\_sequence.nextval, 'Active', 38);

INSERT INTO RESOURCES (resource\_id, resource\_name, resource\_type, description, item\_cost) VALUES (resource\_sequence.nextval, 'MIC', 'demo', 'MIC to demo of project', '123456');

INSERT INTO Activity\_resources (activity\_id, resource\_id, item\_no, item\_required, item\_quantity) VALUES (activity\_sequence.currval, resource\_sequence.currval, 15, 'DIAMOND', '5000');

INSERT INTO COSTS (activity\_id, cost\_id, item\_description, amount) VALUES (activity\_sequence.currval, cost\_sequence.nextval, 'demo slides','5000');

INSERT INTO Reviews (plan\_id, group\_id, review\_id, time, location) VALUES (plan\_sequence.currval, group\_sequence.currval, review\_sequence.nextval,'12NOV18:10:35:57', 'NORTHAMPTON');

INSERT INTO Actions (review\_id, action\_id, owner\_name, deadline) VALUES (review\_sequence.currval, action\_sequence.nextval, 'SIDDHARTH', '22MAR22');

INSERT INTO Activities (activity\_id, name, start\_date) VALUES (activity\_sequence.nextval, 'APP DEVELOPMENT', '07AUG22');

INSERT INTO STAGES (stage\_id, stage\_type, location) VALUES (stage\_sequence.nextval, 'development', 'NEPAL');

INSERT INTO PLANS (activity\_id, stage\_id, plan\_id, start\_date, end\_date) VALUES (activity\_sequence.currval, stage\_sequence.currval, plan\_sequence.nextval, '18AUG21', '21NOV21');

INSERT INTO STAFFS (staff\_id, first\_name, last\_name, contact\_number, email, town, country, date\_registered, gender) VALUES (staff\_sequence.nextval, 'RAJ', 'GURUNG', '9865624725', 'raj@gmail.com', 'KATHMANDU', 'NEPAL', '20APR21', 'M');

INSERT INTO ROLES (role\_id, role\_type, description) VALUES (role\_sequence.nextval, 'Programming', 'To code some program');

INSERT INTO Staff\_roles (staff\_id, role\_id, staff\_role\_id, started\_date, duty\_status) VALUES (staff\_sequence.currval, role\_sequence.currval, s\_r\_sequence.nextval, '15DEC21', 'Active');

INSERT INTO Groups (activity\_id, staff\_role\_id, group\_id, staff\_id, role\_id, status, group\_size) VALUES (activity\_sequence.currval, s\_r\_sequence.currval, group\_sequence.nextval, staff\_sequence.currval, role\_sequence.currval, 'Active', 100);

INSERT INTO RESOURCES (resource\_id, resource\_name, resource\_type, description, item\_cost) VALUES (resource\_sequence.nextval, 'CAM', 'present', 'CAMERA for presentation', '123456');

INSERT INTO Activity\_resources (resource\_id,item\_no, item\_required, item\_quantity) VALUES (resource\_sequence.currval, 11, 'CRYPTO', '10000');

INSERT INTO COSTS (activity\_id, cost\_id, item\_description, amount) VALUES (activity\_sequence.currval, cost\_sequence.nextval, 'pc to manage', '10000');

INSERT INTO Reviews (plan\_id, stage\_id, activity\_id, staff\_role\_id, group\_id, review\_id, time, location) VALUES (plan\_sequence.currval, stage\_sequence.currval, activity\_sequence.currval, s\_r\_sequence.currval, group\_sequence.currval, review\_sequence.nextval,'15OCT22', 'KATHMANDU');

INSERT INTO Actions (action\_id, owner\_name, deadline) VALUES (action\_sequence.nextval, 'AYUSH', '13FEB21');

/\* (Group\_13), 2021 CSY1026 \*/

/\*--- SELECT statement--- \*/

SELECT \* FROM activities;

SELECT \* FROM stages;

SELECT \* FROM plans;

SELECT \* FROM resources;

SELECT \* FROM activity\_resources;

SELECT \* FROM costs;

SELECT \* FROM staffs;

SELECT \* FROM roles;

SELECT \* FROM staff\_roles;

SELECT \* FROM groups;

SELECT \* FROM reviews;

SELECT \* FROM actions;

/\*---- above queries displays all records from given tables--- \*/

SELECT activity\_id, name, start\_date

FROM activities

ORDER BY name DESC;

/\*---it displays activity\_id, name, start\_date from activities

on the basis of descending order of name column----\*/

SELECT stage\_id, stage\_type, location

FROM stages ORDER BY location DESC;

SELECT activity\_id, stage\_ID , START\_DATE

FROM plans

ORDER BY stage\_id DESC;

SELECT resource\_id, resource\_name, resource\_type

FROM resources

ORDER BY resource\_name DESC;

SELECT activity\_id, item\_required, item\_quantity

FROM activity\_resources

ORDER BY item\_required DESC;

SELECT activity\_id, item\_description, amount

FROM costs

ORDER BY item\_description DESC;

SELECT staff\_id, first\_name, last\_name

FROM staffs

ORDER BY first\_name DESC;

SELECT role\_id, role\_type, description

FROM roles

ORDER BY role\_type DESC;

SELECT plan\_id, group\_id, location

FROM reviews

ORDER BY location DESC;

SELECT review\_id, action\_id, owner\_name

FROM actions

ORDER BY owner\_name DESC;

/\*----above queries displays all selected records and displays the given columns records in reverse alphabetical order---\*/

SELECT activity\_id, name

FROM activities

WHERE activity\_id < 5 OR activity\_id >2;

SELECT \*

FROM activities

WHERE activity\_id >= 1 AND activity\_id < 5;

SELECT \* FROM activities

WHERE name BETWEEN 'a%' AND 'z%';

SELECT \* FROM activities WHERE lower(name) NOT BETWEEN 'a%' AND 'hf%';

SELECT \* FROM activities WHERE UPPER(name) LIKE 'A\_%';

SELECT \* FROM activities WHERE name NOT LIKE 'b%';

SELECT \* FROM activities WHERE activity\_id IN(3, 5);

SELECT \* FROM activities WHERE activity\_id NOT IN (3);

SELECT \* FROM stages WHERE stage\_id IN (1, 4);

SELECT \* FROM stages WHERE LOWER(location) = 'northampton';

SELECT \* FROM stages

WHERE LOWER(location) LIKE 'no%';

SELECT \* FROM stages WHERE UPPER(location) <> 'NORTHAMPTON';

SELECT \* FROM stages WHERE LOWER(location) != 'northampton';

SELECT \* FROM stages WHERE UPPER(location) != 'NEPAL';

SELECT stage\_type, location

FROM stages

WHERE LOWER(stage\_type) LIKE '%g';

SELECT plan\_id, activity\_id, stage\_id

FROM plans

WHERE plan\_id > stage\_id;

SELECT \* FROM resources WHERE resource\_name LIKE 'I%';

SELECT \* FROM resources WHERE resource\_name NOT LIKE '%C';

SELECT \* FROM staffs WHERE lower(country) <> 'uk';

SELECT \* FROM staffs WHERE UPPER(town) <> 'KATHMANDU';

SELECT \* FROM staffs WHERE UPPER(country) = 'UK';

SELECT \* FROM staffs WHERE UPPER(town) = 'BEDFORD';

SELECT \* FROM reviews WHERE UPPER(location) = 'NORTHAMPTON';

SELECT \* FROM activities WHERE start\_date BETWEEN '01JAN20' AND '15DEC20';

SELECT \* FROM activities WHERE start\_date > '05JAN19'AND start\_date < '06AUG21';

SELECT \* FROM activities WHERE start\_date > '05JAN19' OR start\_date < '08JUL21';

SELECT \* FROM activities WHERE start\_date > '05JAN20' OR start\_date < '01FEB20';

SELECT plan\_id, end\_date

FROM plans

WHERE start\_date > '01JUL19';

SELECT \* FROM plans WHERE start\_date NOT BETWEEN '07FEB21' AND '05DEC22';

SELECT \* FROM PLANS WHERE end\_date NOT BETWEEN '05JAN21' AND '05JAN22';

/\*--- INNER JOIN QUERIES-----\*/

/\*----- INNER JOIN displays only the record that have matching values of identical columns.----\*/

SELECT Activities.name, Activities.start\_date, Plans.plan\_id, Plans.start\_date

FROM Activities

INNER JOIN Plans

ON Activities.activity\_id = Plans.activity\_id;

SELECT Stages.location, Plans.plan\_id

FROM Stages

INNER JOIN Plans

ON Stages.stage\_id = Plans.stage\_id;

SELECT Plans.plan\_id, Plans.end\_date, Activities.name, Stages.stage\_type

FROM ((Plans

INNER JOIN Activities ON Plans.activity\_id = Activities.activity\_id)

INNER JOIN Stages ON Plans.stage\_id = Stages.stage\_id);

SELECT Activity\_resources.item\_required, Activity\_resources.item\_quantity, Resources.resource\_name, Resources.item\_cost

FROM Activity\_resources

INNER JOIN Resources ON Activity\_resources.resource\_id = Resources.resource\_id;

SELECT Costs.cost\_id, Costs.amount, Activities.name

FROM Costs

INNER JOIN Activities ON Costs.activity\_id = Activities.activity\_id;

SELECT Staff\_roles.staff\_role\_id, Roles.role\_type, Staffs.first\_name, Staffs.last\_name

FROM ((Staff\_roles

INNER JOIN Roles ON Staff\_roles.role\_id = Roles.role\_id)

INNER JOIN Staffs ON Staff\_roles.staff\_id = Staffs.staff\_id);

SELECT Groups.group\_id, Groups.group\_size, Staff\_roles.duty\_status, Activities.name

FROM ((Groups

INNER JOIN Staff\_roles ON Groups.staff\_role\_id = Staff\_roles.staff\_role\_id)

INNER JOIN Activities ON Groups.activity\_id = Activities.activity\_id);

/\*--- LEFT JOIN --- \*/

SELECT Actions.owner\_name, Reviews.location

FROM Actions

LEFT JOIN Reviews ON Actions.review\_id = Reviews.review\_id

ORDER BY owner\_name;

/\* Here LEFT JOIN displays all records from Actions table and only review\_id matching records

from Reviews table \*/

SELECT Activity\_resources.item\_required, Activities.name

FROM Activity\_resources

LEFT JOIN Activities ON Activity\_resources.activity\_id = Activities.activity\_id

ORDER BY item\_required;

SELECT Plans.end\_date, Reviews.review\_id

From Plans

LEFT JOIN Reviews ON Plans.plan\_id = Reviews.plan\_id;

/\* RIGHT JOIN \*/

SELECT Plans.plan\_id, Plans.start\_date, Activities.start\_date

FROM Plans

RIGHT JOIN Activities ON Plans.activity\_id = Activities.activity\_id

ORDER BY plan\_id; /\* Here RIGHT JOIN displays all records of activities and only records having matching column values \*/

SELECT Staff\_roles.duty\_status, Staffs.first\_name, staffs.last\_name

FROM Staff\_roles

RIGHT JOIN staffs ON Staff\_roles.staff\_id = staffs.staff\_id

ORDER BY first\_name;

SELECT Reviews.plan\_id, Reviews.group\_id, Actions.owner\_name , Actions.deadline

FROM Reviews

RIGHT JOIN Actions ON Reviews.review\_id = Actions.review\_id;

/\* FULL OUTER JOIN \*/

SELECT activities.name, plans.plan\_id

FROM activities

FULL OUTER JOIN plans

ON activities.activity\_id = plans.activity\_id;

/\* Here FULL OUTER JOIN displays all the records if there is matchig records or non matching

records between both tables \*/

SELECT plans.plan\_id, stages.stage\_id, stages.stage\_type

FROM plans

FULL OUTER JOIN stages

ON plans.stage\_id = stages.stage\_id

ORDER BY stage\_type;

SELECT staff\_roles.role\_id, staff\_roles.started\_date, staffs.first\_name, staffs.date\_registered

FROM staff\_roles

FULL OUTER JOIN staffs

ON staff\_roles.staff\_id = staffs.staff\_id

ORDER BY first\_name;

/\* EXISTS displays records only if subquery exists \*/

SELECT activity\_id, name

FROM activities

WHERE EXISTS ( SELECT \* FROM stages WHERE stage\_id < 5);

SELECT activity\_id, start\_date

FROM activities

WHERE EXISTS ( SELECT \* FROM plans WHERE plans.activity\_id = activities.activity\_id AND stage\_id > 2);

SELECT stage\_id

FROM stages

WHERE EXISTS ( SELECT \* FROM plans WHERE plans.stage\_id = stages.stage\_id);

SELECT stage\_id

FROM stages

WHERE EXISTS ( SELECT \* FROM plans WHERE stage\_id = activity\_id);

SELECT \* FROM plans

WHERE EXISTS ( SELECT start\_date FROM activities WHERE start\_date NOT LIKE 'SYS%');

/\* ROWNUM select TOP records stored in database \*/

SELECT \* FROM activities

WHERE ROWNUM <=3;

SELECT \* FROM actions

WHERE ROWNUM < 5;

/\* CASE ELSE is like IF THEN statement \*/

SELECT plan\_id, stage\_id,

CASE

WHEN plan\_id > 1000 THEN 'This is new plans'

ELSE 'This is old plans'

END "Plan detail"

FROM plans;

/\* It displays the text 'It is new plans' for records whose plan\_id > 1000

and prints 'This is old plans' for records whose plan\_id is less than 1000 \*/

SELECT stage\_id,

CASE

WHEN location = 'NORTHAMPTON' THEN 'You are in UK'

WHEN location = 'NEPAL' THEN 'You are in NEPAL'

END "location"

FROM stages;

SELECT activity\_id || ' ' || stage\_id "id"

FROM plans;

SELECT first\_name || ' ' || last\_name "name"

FROM staffs

WHERE country = 'UK';

/\* || '' || This joins first\_name and last\_name records from staffs table and displays in new 'name' column \*/

SELECT activity\_id, name,

CASE

WHEN start\_date > '07AUG19' THEN 'This is new'

ELSE ' this is old'

END "description"

FROM activities;

SELECT COUNT (CASE

WHEN activity\_id > 4 THEN activity\_id END) "no of activities"

FROM activities;

SELECT role\_id,

CASE

WHEN UPPER(role\_type) LIKE '%T' THEN 'Your role is project management'

ELSE 'Your role is programming'

END "Roles as"

FROM roles;

SELECT AVG(activity\_id)

FROM activities;

SELECT MIN (stage\_id)

FROM stages;

SELECT MAX (plan\_id)

FROM plans;

SELECT COUNT (\*) FROM activities;

SELECT COUNT (\*) FROM reviews;

SELECT COUNT (\*) FROM staffs;

/\* COUNT (\*) displays no of rows from given table \*/

/\* To view all CONSTRAINT names \*/

SELECT CONSTRAINT\_NAME

FROM USER\_CONSTRAINTS

WHERE CONSTRAINT\_NAME NOT LIKE 'SYS%';

/\* To view all TABLE names \*/

SELECT TABLE\_NAME

FROM USER\_TABLES

WHERE TABLE\_NAME NOT LIKE 'SYS%';

/\* GROUP BY\*/

/\* GROUP BY is used to aggregate functions \*/

/\* COUNT() , MIN(), MAX(), SUM(), AVG(), etc can be used with GROUP BY \*/

SELECT COUNT(stage\_id) "no of stages" , location

FROM stages

GROUP BY location; /\* Here it displays the no of stages in each location \*/

SELECT COUNT(stage\_id) "no of stages" , location

FROM stages

GROUP BY location

HAVING COUNT (location) > 1;

/\* It displayes the no of stages which is more than one in same location \*/

SELECT COUNT(stage\_id) "no of stages" , location

FROM stages

GROUP BY location

HAVING COUNT (stage\_id) > 1;

/\* It displays more than one stages in same location \*/

SELECT COUNT(stage\_id) "no of stages" , location

FROM stages

GROUP BY location

HAVING MIN(stage\_id) > 1;

/\* It displays the no of stages whose minimum stage id value is greater than 1000 \*/

SELECT COUNT(stage\_id) "no of stages" , location

FROM stages

GROUP BY location

HAVING AVG(stage\_id) > 2;

/\* ANY ALL \*/

/\*ANY returns true if any subquery meets the condition and displays the records \*/

/\*ALL returns true if all subquery meets the condition \*/

SELECT stage\_id

FROM plans

WHERE stage\_id = ANY(SELECT stage\_id FROM stages WHERE location = 'NORTHAMPTON' OR LOCATION ='UK');

/ \* here if any records of plans table has stage\_id whose location = 'NORTHAMPTON' OR 'UK', it displays stage\_id\*/

SELECT stage\_id

FROM plans

WHERE stage\_id = ALL(SELECT stage\_id FROM stages WHERE location = 'NEPAL' AND LOCATION = 'NORTHAMPTON');

/\* If records of plans table have must have stage\_id whose location = 'NEPAL' AND 'NORTHAMPTON' then only it displays the stage\_id \*/

SELECT \* FROM STAGES

WHERE location = ANY('NORTHAMPTON', 'NEPAL');

SELECT \* FROM STAGES

WHERE location = ALL('NORTHAMPTON');

/\* (Group\_13), 2021 CSY1026 \*/

/\*----------DROP A FOREIGN KEY CONSTRANT------------\*/

ALTER TABLE actions

DROP CONSTRAINT fk\_a\_reviews;

ALTER TABLE reviews

DROP CONSTRAINT fk\_r\_groups;

ALTER TABLE reviews

DROP CONSTRAINT fk\_r\_plans;

ALTER TABLE groups

DROP CONSTRAINT fk\_g\_staff\_roles;

ALTER TABLE groups

DROP CONSTRAINT fk\_g\_activities;

ALTER TABLE staff\_roles

DROP CONSTRAINT fk\_s\_r\_staffs;

ALTER TABLE staff\_roles

DROP CONSTRAINT fk\_s\_r\_roles;

ALTER TABLE costs

DROP CONSTRAINT fk\_c\_activities;

ALTER TABLE activity\_resources

DROP CONSTRAINT fk\_a\_r\_activities;

ALTER TABLE activity\_resources

DROP CONSTRAINT fk\_a\_r\_resources;

ALTER TABLE plans

DROP CONSTRAINT fk\_p\_activities;

ALTER TABLE plans

DROP CONSTRAINT fk\_p\_stages;

/\*----------DROP PRIMARY KEY CONSTRANT------------\*/

ALTER TABLE actions

DROP CONSTRAINT pk\_actions;

ALTER TABLE reviews

DROP CONSTRAINT pk\_reviews;

ALTER TABLE groups

DROP CONSTRAINT pk\_groups;

ALTER TABLE staff\_roles

DROP CONSTRAINT pk\_staff\_roles;

ALTER TABLE roles

DROP CONSTRAINT pk\_roles;

ALTER TABLE staffs

DROP CONSTRAINT pk\_staffs;

ALTER TABLE costs

DROP CONSTRAINT pk\_costs;

ALTER TABLE activity\_resources

DROP CONSTRAINT pk\_activity\_resources;

ALTER TABLE resources

DROP CONSTRAINT pk\_resources;

ALTER TABLE plans

DROP CONSTRAINT pk\_plans;

ALTER TABLE stages

DROP CONSTRAINT pk\_stages;

ALTER TABLE activities

DROP CONSTRAINT pk\_activities;

/\*----------DROP DEFAULT------------\*/

/\*---DROPPING DEFAULT IS NOT GOOD IDEA SO SET IT TO NULL -----\*/

ALTER TABLE activities

MODIFY start\_date DEFAULT NULL;

ALTER TABLE Stages

MODIFY location DEFAULT NULL;

ALTER TABLE Staffs

MODIFY town DEFAULT NULL;

ALTER TABLE Staffs

MODIFY country DEFAULT NULL;

ALTER TABLE Staffs

MODIFY date\_registered DEFAULT NULL;

ALTER TABLE activities

MODIFY start\_date DEFAULT NULL;

/\*----------DROP A UNIQUE CONSTRAINT------------\*/

ALTER TABLE resources

DROP CONSTRAINT u\_resources;

ALTER TABLE staffs

DROP CONSTRAINT u\_staffs;

ALTER TABLE roles

DROP CONSTRAINT u\_roles;

/\*---- TO DROP CHECK CONSTRAINT ----\*/

ALTER TABLE activities

DROP CONSTRAINT ck\_activities\_name\_upper;

ALTER TABLE resources

DROP CONSTRAINT ck\_resources\_name\_upper;

ALTER TABLE staffs

DROP CONSTRAINT ck\_staffs\_first\_name\_upper;

ALTER TABLE staffs

DROP CONSTRAINT ck\_staffs\_last\_name\_upper;

ALTER TABLE staffs

DROP CONSTRAINT ck\_staffs\_gender;

ALTER TABLE staff\_roles

DROP CONSTRAINT ck\_staff\_roles\_duty\_status;

ALTER TABLE groups

DROP CONSTRAINT ck\_groups\_status;

ALTER TABLE actions

DROP CONSTRAINT ck\_actions\_owner\_name\_upper;

/\*----drop sequence----\*/

DROP SEQUENCE activity\_sequence;

DROP SEQUENCE stage\_sequence;

DROP SEQUENCE plan\_sequence;

DROP SEQUENCE resource\_sequence;

DROP SEQUENCE a\_r\_sequence;

DROP SEQUENCE cost\_sequence;

DROP SEQUENCE staff\_sequence;

DROP SEQUENCE role\_sequence;

DROP SEQUENCE s\_r\_sequence;

DROP SEQUENCE group\_sequence;

DROP SEQUENCE review\_sequence;

DROP SEQUENCE action\_sequence;

/\*----------DROP TABLE ACTIVITIES------------\*/

/\* PURGE doesnot send these tables to RECYCLEBIN \*/

DROP TABLE Actions PURGE;

DROP TABLE Reviews PURGE;

DROP TABLE groups PURGE;

DROP TABLE staff\_roles PURGE;

DROP TABLE roles PURGE;

DROP TABLE staffs PURGE;

DROP TABLE costs PURGE;

DROP TABLE Activity\_resources PURGE;

DROP TABLE resources;

DROP TABLE plans;

DROP TABLE stages;

DROP TABLE activities;

/\* Without droping each constraint we can directly delete table

by using CASCADE CONSTRAINT but it violates the rule of referential integrity so it

is not good pactice

Eg:

DROP TABLE activities CASCADE CONSTRAINT;

DROP TABLE plans CASCADE CONSTRAINT;

\*/

/\* use \*/

PURGE RECYCLEBIN;

/\* to empty RecycleBin \*/

/\* (Group\_13), 2021 CSY1026 \*/

/\*---run cmd To create tables --- \*/

@D:\dbassignment2\create\_13.txt

/\*---run cmd to alter tables ----\*/

@D:\dbassignment2\alter\_13.txt

/\*-----run cmd to insert values in tables --- \*/

@D:\dbassignment2\insert\_13.txt

/\*---run cmd for SELECT queries --- \*/

@D:\dbassignment2\queries\_13.txt

/\*--- run cmd to drop constraints, sequence, default, tables ---\*/

@D:\dbassignment2\drop\_13.txt