Lab 05- DDL

Objective:

The purpose of this lab is to introduce you to the DDL set of statements in SQL. By writing SQL to create tables, constraints, and views, you will have the tools needed to implement database designs that you will create later in the course. By finishing this lab, the student will be able to:

- create, modify, and drop tables based on design specifications provided,
- inserting new data into tables, update data in tables, and delete data from tables while considering referential integrity,
- enforce constraints on tables to ensure data integrity and consistency,
- create a table using the structure and data from an existing table,
- import data into a table from other tables.

Submission:

Your submission will be a single WORD file with the solutions provided.

Your submission needs to include a comment header block and be commented to include the question and the solutions. Make sure every SQL statement terminates with a semicolon.

Tasks:

Add
`SET AUTOCOMMIT ON;`
under the comment header and execute it

Consider the following table specifications:

Part A (DDL) :

1. Create all the following tables and their given constraints:

MOVIES `(movieid:int, title:varchar(35), releaseYear:int, director:int,score:decimal(3,2))`

Column Name	Column DataType	PK	Not Null	Unique	FK	Default Value	Validation
movieid	Int	✓					
title	varchar(35)		✓				
releaseYear	Int		✓				
director	Int		✓				
score	decimal(3,2)						< 10 and > 3

```
CREATE TABLE MOVIES (
movieid INT PRIMARY KEY,
title VARCHAR(35) NOT NULL,
releaseYear INT NOT NULL,
director INT NOT NULL,
score DECIMAL(3,2),
CONSTRAINT score CHECK  (score BETWEEN 3 AND 10)
);
```

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE
1	MOVIEID	NUMBER(38,0)	No
2	TITLE	VARCHAR2(35 BYTE)	No
3	RELEASEYEAR	NUMBER(38,0)	No
4	DIRECTOR	NUMBER(38,0)	No
5	SCORE	NUMBER(3,2)	Yes

ACTORS `(actorid:int, firstname:varchar(20), lastname:varchar(30))`

Column Name	Column DataType	PK	Not Null	Unique	FK	Default Value	Validation
actorid	Int	✓					
firstName	varchar(20)		✓				
lastName	Varchar(30)		✓				

```
CREATE TABLE ACTORS (
actorid INT PRIMARY KEY,
firstName VARCHAR(20) NOT NULL,
lastName VARCHAR(30) NOT NULL
);
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DA
1	ACTORID	NUMBER(38,0)	No	(
2	FIRSTNAME	VARCHAR2(20 BYTE)	No	(
3	LASTNAME	VARCHAR2(30 BYTE)	No	(

CASTINGS (movieid:int, actorid:int)

Column Name	Column DataType	PK	Not Null	Unique	FK	Default Value	Validation
movieid	Int	✓			✓ (movies)		
actorid	int	✓			✓ (actors)		

```
CREATE TABLE CASTINGS (
movieid INT,

actorid INT,

CONSTRAINT keys_primaryfields PRIMARY
KEY(movieid,actorid),

ADD CONSTRAINT movies_fk FOREIGN KEY (movieid) REFERENCES MOVIES(movieid),
ADD CONSTRAINT actor_fk FOREIGN KEY (actorid) REFERENCES ACTORS(actorid)
);
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE
1	MOVIEID	NUMBER(38,0)
2	ACTORID	NUMBER(38,0)

DIRECTORS (directorid:int, firstname:varchar(20), lastname:varchar(30))

Column Name	Column DataType	PK	Not Null	Unique	FK	Default Value	Validation
directorid	Int	✓					
firstname	varchar(20)		✓				
lastname	varchar(30)		✓				

```
CREATE TABLE DIRECTORS (
directorid INT PRIMARY KEY,
firstName VARCHAR(20) NOT NULL,
lastName VARCHAR(30) NOT NULL
);
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLAB
1	DIRECTORID	NUMBER(38,0)	No
2	FIRSTNAME	VARCHAR2(20 BYTE)	No
3	LASTNAME	VARCHAR2(30 BYTE)	No

2. Modify the ***movies*** table to create a foreign key constraint that refers to table ***directors***.

```
ALTER TABLE MOVIES

ADD CONSTRAINT movies_director_fk FOREIGN KEY (director) REFERENCES
DIRECTORS(directorid);
```

Table MOVIES altered.

3. Modify the ***movies*** table to create a new constraint so the uniqueness of the movie title is guaranteed.
   ALTER TABLE MOVIES
   ADD CONSTRAINT u_title UNIQUE (title);

4. Write insert statements to add the following data to table ***directors*** and ***movies***.

Director

directorid	First name	Last name
1010	Rob	Minkoff
1020	Bill	Condon
1050	Josh	Cooley
2010	Brad	Bird
3020	Lake	Bell

INSERT ALL

INTO DIRECTORS VALUES(1010, 'Rob', 'Minkoff') INTO

DIRECTORS VALUES(1020, 'Bill', 'Condon') INTO

DIRECTORS VALUES(1050, 'Josh', 'Cooley') INTO

DIRECTORS VALUES(2010, 'Brad', 'Bird') INTO

DIRECTORS VALUES(3020, 'Lake', 'Bell') SELECT * FROM

DUAL;

COMMIT

Movies

Id	title	year	director	score
100	The Lion King	2019	3020	3.50
200	Beauty and the Beast	2017	1050	4.20
300	Toy Story 4	2019	1020	4.50
400	Mission Impossible	2018	2010	5.00
500	The Secret Life of Pets	2016	1010	3.90

INSERT INTO MOVIES (movieid, title, releaseyear, director, score) VALUES (100, 'The Lion King', 2019, 3020, 3.50);
INSERT INTO MOVIES (movieid, title, releaseyear, director, score) VALUES (200, 'Beauty and the Beast', 2017, 1050, 4.20);
INSERT INTO MOVIES (movieid, title, releaseyear, director, score) VALUES (300, 'Toy Story 4', 2019, 1020, 4.50);
INSERT INTO MOVIES (movieid, title, releaseyear, director, score) VALUES (400, 'Mission Impossible', 2018, 3020, 5.00);
INSERT INTO MOVIES (movieid, title, releaseyear, director, score) VALUES (500, 'The Secret Life of Pets', 2016, 1010, 3.90);

commit;

5. Write SQL statements to remove all above tables.
   Is the order of tables important when removing? Why?
   DROP TABLE CASTINGS; DROP TABLE MOVIES; DROP TABLE ACTORS; DROP TABLE DIRECTORS;
   Table CASTINGS dropped.
   Table MOVIES dropped. Table ACTORS dropped.

Table DIRECTORS dropped.

Child tables must be dropped first before dropping parent table. You can't drop a parent table if you have a child table with a foreign key constraint in place, unless you specify the CASCADE CONSTRAINTS clause: DROP TABLE P CASCADE CONSTRAINTS; This command drops the FK constraint too. Deleting a table will necessarily drop all constraints related to this table.

DROP TABLE MOVIES CASCADE CONSTRAINTS; DROP
TABLE ACTORS CASCADE CONSTRAINTS;

DROP TABLE CASTINGS CASCADE CONSTRAINTS;

DROP TABLE DIRECTORS CASCADE CONSTRAINTS;
Table CASTINGS dropped.
Table MOVIES dropped. Table
ACTORS dropped. Table
DIRECTORS dropped.

Part B (More DML):

6. Create a new empty table (that means the table will not have any data after creating) ***employeecopy*** the same as table ***retailemployees.*** Use a single statement to create the table and insert the data at the same time (Hint use a WHERE clause that is false like 1=2)
   CREATE TABLE employeecopy AS SELECT * FROM employees where 1=2;

   Table EMPLOYEECOPY created.

7. Modify table ***employeecopy*** and add a new column ***username*** to this table. The value of this column is not required and does not have to be unique.

   ALTER TABLE employeeCOPY

   ADD username VARCHAR(10);

   Table EMPLOYEESCOPY altered.

8. Re-insert all data from the ***retailemployees.*** table into your new table ***employeecopy*** using a single statement.
   INSERT INTO employeecopy (EMPLOYEENUMBER , LASTNAME ,FIRSTNAME ,EXTENSION , EMAIL ,OFFICECODE ,REPORTSTO ,JOBTITLE)
   SELECT EMPLOYEENUMBER ,LASTNAME ,FIRSTNAME ,EXTENSION ,EMAIL , OFFICECODE ,REPORTSTO ,JOBTITLE FROM retailemployees;
   23 rows inserted.
9. In table ***employeecopy***, generate the email address for column ***username*** for each student by concatenating the employeeid and the string "@seneca.ca". For instance, the username of employee 123 will be "123@seneca.ca'.
   SELECT employeenumber || '@seneca.ca' AS username

FROM employeecopy;

10. Delete all the employeecopy data and display the data in the table. Does employeecopy exist? If not how can you delete table ***employeecopy***.
    DELETE FROM employeecopy;
    Yes employeecopy structure still exist even when the data is deleted. DROP

    TABLE employeecopy;

11. Create a statement that will insert yourself as an RETAILEMPLOYEE of the company.
    - Use a unique employee number of your choice
    - Use your school email address
    - Your job title will be "Cashier"
    - Office code will be 4
    - You will report to employee 1088

    ```
    INSERT INTO
    ```

```
RETAILEMPLOYEES(employeeNumber,lastName,firstName,extension,email,officeCode,re
portsTo,jobTitle)
values(55555,'GNA','rr',152,'r@senecacollege.ca',4,1088,'cashier')
```

	EMPLOYEENUMBER	LASTNAME	FIRSTNAME	EXTENSION	EMAIL	OFFICECODE	REPORT
1	55555	GNA	rr	152	r@senecacolle...	4	1(

```
--1 row inserted with above info
```

12. Create a query that displays your, and only your, RETAILEMPLOYEE data

```
SELECT *
FROM RETAILEMPLOYEES
WHERE employeeNumber=55555
--new info selected
```

	EMPLOYEENUMBER	LASTNAME	FIRSTNAME	EXTENSION	EMAIL	OFFICECODE	REPORTSTO	JOBTITLE
1	55555	GNA	rr	152	r@senecacollege.ca	4	1088	cashier

13. Create a statement to update your job title to “Head Cashier”

```
UPDATE RETAILEMPLOYEES SET jobTitle='Head Cashier'
WHERE employeeNumber=55555

SELECT *
FROM RETAILEMPLOYEES WHERE employeeNumber=55555
```

	EMPLOYEENUMBER	LASTNAME	FIRSTNAME	EXTENSION	EMAIL	OFFICECODE	REPORTSTO	JOBTITLE
1	55555	GNA	rr	152	r@senecacollege.ca	4	1088	Head Cashier

14. Create a statement to insert another fictional RETAILEMPLOYEE into the database. This RETAILEMPLOYEE will be a “Cashier” and will report to you. Make up fake data for the other fields.

```
INSERT INTO
RETAILEMPLOYEES
(employeeNumber,lastName,firstName,extension,email,officeCode,reportsTo,job
Title)
values(65555,'GNAN','rRr',152,'r@senecacollege.ca',4,55555,'cashier')

SELECT *
FROM RETAILEMPLOYEES
WHERE employeeNumber=65555
```

	EMPLOYEENUMBER	LASTNAME	FIRSTNAME	EXTENSION	EMAIL	OFFICECODE	REPORTSTO	JOBTITLE
1	65555	GNAN	rRr	152	r@senecacollege.ca	4	55555	cashier

15. Create a statement to Delete yourself from the database. Did it work? If not, why?

```
DELETE
FROM RETAILEMPLOYEES
WHERE employeeNumber=55555

--REPORTSTO FOREIGN KEY CONFLICT
```

```
FROM RETAILEMPLOYEES
WHERE employeeNumber=55555
Error report -
ORA-02292: integrity constraint (RGNANAOLIVU.EMP_RTEMP_FK) violated - child record found
```

16. Create a statement to delete the fake employee from the database and then rerun the statement to delete yourself. Did it work?

```
DELETE
FROM RETAILEMPLOYEES
WHERE employeeNumber=65555

--1 ROW DELETED

DELETE
FROM RETAILEMPLOYEES WHERE employeeNumber=55555
       --NOW it deleted my record because FOREIGN KEY RECORD IS DELETED
```

17. Create a **single** statement that will insert both yourself and the fake employee at the same time. This time the fake employee will report to 1088 as well.

```
INSERT ALL
INTO RETAILEMPLOYEES
(employeeNumber,lastName,firstName,extension,email,officeCode,reportsTo,job
Title)
values(55555,'GNA','rr',152,'r@senecacollege.ca',4,1088,'cashier')
INTO RETAILEMPLOYEES
(employeeNumber,lastName,firstName,extension,email,officeCode,reportsTo,job
Title)
values(65555,'GNAN','rRr',152,'r@senecacollege.ca',4,1088,'cashier')
SELECT * FROM DUAL;

--2 ROWS INSERTED
```

18. Create a **single** statement to delete both yourself and the fake employee.

```
DELETE
FROM RETAILEMPLOYEES
WHERE employeeNumber=55555 OR employeeNumber=65555

--2 ROWS DELETED
```

19. Create a new order in RETAILORDER table with required date Sep 22nd,2021 and order date as Sep 17th,2021. Make up the reset of the fields and then display the only the new order that you have created just now.

```
INSERT INTO
RETAILORDERS(orderNumber,orderDate,requiredDate,shippedDate,status,comments,custom
erNumber)
VALUES(655555,to_date('2021-09-17','yyyy-mm-dd'),to_date('2021-09-21','yyyy-mm-
dd'),to_date('2021-09-20','yyyy-mm-dd'),'Shipped','order created',103)
COMMIT;
       --1 ROW INSERTED
```

Commit complete.

	ORDERNUMBER	ORDERDATE	REQUIREDDATE	SHIPPEDDATE	STATUS	COMMENTS
1	10422	30-MAY-05	11-JUN-05	(null)	In Process	(null)
2	10423	30-MAY-05	05-JUN-05	(null)	In Process	(null)
3	10424	31-MAY-05	08-JUN-05	(null)	In Process	(null)
4	10425	31-MAY-05	07-JUN-05	(null)	In Process	(null)
5	655555	17-SEP-21	21-SEP-21	20-SEP-21	Shipped	order create
6	10100	06-JAN-03	13-JAN-03	10-JAN-03	Shipped	(null)

20. Insert a new product into product table with product name as “2020 Bugatti Veyron” and productcode as “S111_111” and make up the rest of the fields.

```
INSERT INTO

RETAILPRODUCTS(productCode,productName,productLine,productScale,productVendor,prod
uctDescription,quantityInStock,buyPrice,MSRP)

VALUES('S111_111','2020 Bugatti Veyron','CLASSIC CARS','1:10','Second Gear
Diecast','This beast of an engine employs four turbochargers to generate a mighty
1500 horsepower',1,119,120)
       --1 ROW INSERTED
```

31	S72_3212	Pont Yacht	Ships	1:72	U
32	S111_111	2020 Bugatti Veyron	Classic Cars	1:10	S
33	S10_1678	1969 Harley Davids...	Motorcycles	1:10	M