Ex.no: 01

Date:

DDL, DML COMMANDS AND INTEGRITY CONSTRAINTS

AIM:

To perform DDL commands like create, alter, rename, truncate, drop and DML commands like insert, update, select, delete and DDL with integrity constraints for University Database.

DDL COMMANDS:

1. Create
2. Alter
3. Rename
4. Truncate
5. Drop

1.CREATE

SQL> **create table uni_teacher(id int,**

name char(20),

salary int, session varchar(20));

Table created.

SQL> **desc uni_teacher;**

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
name	char(20)	YES		NULL	
amount	int	YES		NULL	
position	char(20)	YES		NULL	
session	varchar(20)	YES		NULL	

SQL> insert into uni_teacher values (101, 'siva', 10000, 'teacher');

insert into uni_teacher values(106, 'siva', 20000,'placement_incharge','fn');

insert into uni_teacher values(103, 'ashwath', 20000, 'teacher',"");

insert into uni_teacher values(104, 'sumathi', 40000, 'teacher');

SQL> select * from uni_teacher;

id	name	amount	position	session
106	siva	20000	placement_incharge	fn
103	ashwath	20000	teacher	
105	jaan	35000	pt_incharge	
104	sumathi	40000	teacher	

2.ALTER

i) To add field name as salary

SQL> alter table uni_teacher change amount salary int;
SQL> select * from uni_teacher;

id	name	salary	position	session
106	siva	20000	placement_incharge	fn
103	ashwath	20000	teacher	
105	jaan	35000	pt_incharge	
104	sumathi	40000	teacher	

ii) To modify salary field type:

SQL> alter table uni_teacher modify salary bigint;

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
name	char(20)	YES		NULL	
salary	bigint	YES		NULL	
position	char(20)	YES		NULL	
session	varchar(20)	YES		NULL	

iii) To rename field name amount to salary:

SQL> alter table uni_teacher rename column amount to salary;

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
name	char(20)	YES		NULL	
salary	bigint	YES		NULL	
position	char(20)	YES		NULL	
session	varchar(20)	YES		NULL	

iv) To drop field name:

SQL> alter table uni_teacher drop column session;

id	name	salary	position
106	siva	20000	placement_incharge
103	ashwa	20000	teacher
105	jaan	35000	pt_incharge
104	sumathi	40000	teacher

3.RENAME

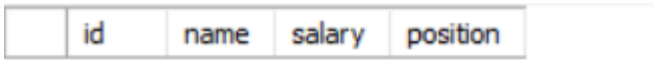
SQL> alter table uni_teacher rename to uni_teachers;

desc uni_teachers;

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
name	char(20)	YES		NULL	
salary	bigint	YES		NULL	
position	char(20)	YES		NULL	

4.TRUNCATE

SQL>
truncate table uni_teachers;



5.DROP

SQL> drop table uni_teachers;
Table dropped.

ERROR:
ORA-04043: object uni_teachers does not exist

DML COMMANDS

1.Insert

2.Update

3.Select

4.Delete

1.Insert command

SQL>insert into uni_nonteaching
(id,name,age,salary,dpt_id)values(101,'siva',20,15500,10001),(102,'sakthi',26,20000,10001)
,(103,'ram',34,25000,10003),(104,'jack',21,18000,10004);

2.Select Command

SQL> select * from uni_nonteaching;

id	name	age	salary	dpt_id
101	siva	20	15500	10001
102	sakthi	26	20000	10001
103	ram	34	25000	10003
104	jack	21	18000	10004

jack

3.Update Command

update uni_nonteaching set salary=20000 where id =101;

id	name	age	salary	dpt_id
101	siva	20	20000	10001
102	sakthi	26	20000	10001
103	ram	34	25000	10003
104	jack	21	18000	10004

4.Delete command

delete from uni_nonteaching where id=104;

1 row deleted.

select * from uni_nonteaching;

id	name	age	salary	dpt_id
101	siva	20	20000	10001
102	sakthi	26	20000	10001
103	ram	34	25000	10003

26

INTEGRITY CONSTRAINTS

1.Primary Key

2.Foreign Key

3.Unique Key

4.Check constraint

5.Not null constraint

6.Null constraint

Creating the Student Table with Integrity Constraints

```
create table uni_teacher( id int primary key,
name char(20) not null,
salary int,
session varchar(10) default 'FN'
);
```

Inserting Values into the uni_teacher Table

```
insert into uni_teacher (id, name, salary, session) values
(101,'siva',20000,default),(102,'vivin',19000,default),(103,'meena',30000,default
);
```

id	name	salary	session
101	siva	20000	FN
102	vivin	19000	FN
103	meena	30000	FN
NULL	NULL	NULL	NULL

Creating the uni_nonteaching Table with Foreign Key

```
create table uni_nonteaching(id int, name char(20),age int,salary int,dpt_id
int,constraint f1 foreign key(dpt_id) references uni_dpt(dpt_id));
```

Inserting Values into the Courses Table

insert into uni_nonteaching
(id,name,age,salary,dpt_id)values(101,'siva',20,15500,10001),(102,'sakthi',26,20000,10001),(103,'ram',34,25000,10003),(104,'jack',21,18000,10004);

id	name	age	salary	dpt_id
101	siva	20	20000	10001
102	sakthi	26	20000	10001
103	ram	34	25000	10003

COE (30)	
RECORD(20)	
VIVA (10)	
TOTAL (60)	

RESULT:

Thus, the execution of DDL, DML commands and integrity constraints for University Database system has been done successfully.

Ex.no: 02

Date:

TCL COMMANDS AND DCL COMMANDS

AIM:

To perform transaction control language commands like savepoint, rollback, commit and data control language commands like grant, revoke for University Database system.

TCL COMMANDS:

1. Savepoint
2. Rollback
3. Commit

Table creation:

```
create table uni_teacher( id int primary key,
name char(20) not null,
salary int,
session varchar(10) default 'FN'
);
```

Table created.

insert into uni_teacher (id, name, salary, session) values
(101,'siva',20000,default),(102,'vivin',19000,default),(103,'meena',30000,default);

id	name	salary	session
101	siva	20000	FN
102	vivin	19000	FN
103	meena	30000	FN
NULL	NULL	NULL	NULL

Creating a Savepoint:

1.Savepoint

start transaction;

savepoint s1;

insert into uni_teacher values (104,'loki',20000,default);

savepoint s2;

update uni_teacher set session='AN' where id=104;

select * from uni_teacher;

id	name	salary	session
101	siva	20000	FN
102	vivin	19000	FN
103	meena	30000	FN
104	loki	200 30000	
NULL	NULL	NULL	NULL

2.Rollback

SQL> rollback to s2;

Rollback complete.

SQL>Select *from student;

id	name	salary	session
101	siva	20000	FN
102	vivin	19000	FN
103	meena	30000	FN
104	loki	20000	FN
NULL	NULL	NULL	NULL

Notice that **session** has been updated.

3.Commit

COMMIT;

Commit complete.

SQL> rollback to s1;

ORA-01086: savepoint 'S1' never established in this session or is invalid.

DCL COMMANDS:

1. Grant
2. Revoke

1.Creating Users:

SQL>create user student identified by 'student';

User created.

2. Granting Privileges to Users:

SQL> grant select on uni_teacher to student;

Grant succeeded.

3.Nisanth Logs In and SELECTs Data:

Select*from uni_teacher;

id	name	salary	session
101	siva	20000	FN
102	vivin	19000	FN
103	meena	30000	FN
104	loki	20000	FN
NULL	NULL	NULL	NULL

select * from uni_nonteaching;

12:54:45 select * from uni_nonteaching LIMIT 0, 1000 Error Code: 1142. SELECT command denied to user 'student'@'localhost' for table 'uni_nonteaching' 0.000 sec

4. Revoking Privileges:

revoke all on uni_teacher from student;

Revoke succeeded.

Trying to INSERT after REVOKE*:*

select * from uni_teacher;

ERROR: ORA-01031: insufficient privileges
12:56:39 select * from uni_teacher LIMIT 0, 1000 Error Code: 1142. SELECT command denied to user 'student'@'localhost' for table 'uni_teacher' 0.000 sec

5. DROP USER

drop user student;

12:57:39 drop user student 0 row(s) affected 0.000 sec

COE (30)	
RECORD (20)	
VIVA (10)	
TOTAL (60)	

RESULT:

Thus, the execution of TCL, DCL commands for University Database system has been done successfully.

Ex.no: 03

Date:

DATABASE OBJECTS

AIM:

To perform database objects such as view, sequences, synonyms and indexes for University Database System.

DATABASE OBJECTS:

1. View
2. Sequences
3. Synonyms
4. Indexes

1.View

Table creation:

create table uni_teacher(id int primary key,

name char(20) not null, salary int, session varchar(10) default 'FN');

Table created.

insert into uni_teacher (id, name, salary, session) values
(101,'siva',20000,default),(102,'vivin',19000,default),(103,'meena',30000,default);

id	name	salary	session
101	siva	20000	FN
102	vivin	19000	FN
103	meena	30000	FN
104	loki	20000	FN

1. Creating Read-Only Views:

SQL>create view v1 as select * from uni_teacher;

create user student identified by 'student';

grant select on v1 to student;

select * from v1;

View created.

SQL> select * from v1;

id	name	salary	session
101	siva	20000	FN
102	vivin	19000	FN
103	meena	30000	FN
104	loki	20000	FN

SQL> insert into v1 values(105, 'lufi',30000,default);

19:02:17 insert into v1 values(105, 'lufi',30000,default) Error Code: 1142. INSERT command denied to user 'student'@'localhost' for table 'v1'0.000 sec

2. Creating Updateable Views:

SQL> create view v2 as select * from uni_teacher;

grant select, update on v2 to student;

View created.

update v2 set session ='AN' where id=104;

SQL> select * from v2;

id	name	salary	session
101	siva	20000	FN
102	vivin	19000	FN
103	meena	30000	FN
104	loki	20000	AN

3. Working with Sequences:

SQL> create sequence seq1 increment by 1 start with 6;

Sequence created.

SQL> insert into student values (seq1.nextval, 'Vasanth', 'MCA', 'vasanth.24mca@kongu.edu', 9994175646);

1 row created.

SQL> select * from student;

SID	SNAME	DPMT	EMAIL	CONTACTNO
101	Nisanth	MCA	nisanthg.24mca@kongu.edu	9345292781
102	Sivakumar	MCA	sivakumarp.24mca@kongu.edu	8072363074
103	Sachin	MCA	sachins.24mca@kongu.edu	8754681258
104	Siva	MCA	sivak.24mca@kongu.edu	6380603146
105	Sujith	MCA	sujith,24mca@kongu.edu	7305207628
106	Vasanth	MCA	vasanth.24mca@kongu.edu	9994175646

4. Synonyms:

SQL> create synonym stud_syn for student;

Synonym created.

SQL> select * from stud_syn;

SID	SNAME	DPMT	EMAIL	CONTACTNO
101	Nisanth	MCA	nisanthg.24mca@kongu.edu	9345292781
102	Sivakumar	MCA	sivakumarp.24mca@kongu.edu	8072363074
103	Sachin	MCA	sachins.24mca@kongu.edu	8754681258
104	Siva	MCA	sivak.24mca@kongu.edu	6380603146

5. Indexes:

SQL> create index idx_sid on student(sid);

Index created.

This creates an index named idx_sid on the sid column of the student table. Indexes help to speed up the retrieval of rows by providing quick lookups on specific columns.

SQL> drop index idx_sid;

Index dropped.

This removes the index idx_sid from the student table, meaning the database will no longer use this index for queries involving the sid column.

COE (30)	
RECORD (20)	
VIVA (10)	
TOTAL (60)	

RESULT:

Thus, the execution of database objects for University Database System has been done successfully.

Ex.no: 04

Date:

SINGLE ROW FUNCTIONS, AGGREGATE FUNCTIONS AND SET OPERATIONS

AIM:

To implement single row functions, aggregate functions and set operations for University Database system.

SINGLE ROW FUNCTIONS:

1. General functions
2. Case conversion functions
3. Character functions
4. Date functions
5. Number functions

Table creation:

```
create table student (
   sid int(10),
   sname varchar(20),
   dpmt varchar(5),
   email varchar(25),
   contactno numeric(10)
);
```

Table created.

SQL> **insert into student values**

(101, 'Nisanth', 'MCA', 'nisanthg.24mca@kongu.edu', 9345292781),

(102, 'Sivakumar', 'MCA', 'sivakumarp.24mca@kongu.edu', 8072363074),

(103, 'Sachin', 'MCA', 'sachins.24mca@kongu.edu', 8754681258),

(104, 'Siva', 'MCA', 'sivak.24mca@kongu.edu', 6380603146);

SQL> select * from student;

SID	SNAME	DPMT	EMAIL	CONTACTNO
101	Nisanth	MCA	nisanthg.24mca@kongu.edu	9345292781
102	Sivakumar	MCA	sivakumarp.24mca@kongu.edu	8072363074
103	Sachin	MCA	sachins.24mca@kongu.edu	8754681258
104	Siva	MCA	sivak.24mca@kongu.edu	6380603146

1. General Functions

a) NVL () - Replace NULL with a default value:

SQL> select sid, nvl (contactno, 'no contact') as contact from student;

SID	CONTACTNO
101	9345292781
102	8072363074
103	8754681258
104	6380603146

Insert with NULL for Contact Number:

SQL> insert into student (sid, sname, dpmt, email, contactno) values (105, 'vasanth', 'mca', 'vasanthv.24mca@kongu.edu', null);

Query to Display Contact with NVL():

SQL> select sid, sname, nvl(contactno, 'no contact') as contact
from student;

SID	SNAME	CONTACTNO
101	Nisanth	9345292781
102	Sivakumar	8072363074
103	Sachin	8754681258
104	Siva	6380603146
105	Vasanth	**No contact**

b) DECODE () - Conditional replacement:

SQL> select sid, sname, decode(dpmt, 'mca', 'master of computer applications', 'unknown') as department from student;

SID	SNAME	DPMT
101	Nisanth	Master of ComputerApplication
102	Sivakumar	Master of Computer Application
103	Sachin	Master of Computer Application
104	Siva	Master of Computer Application

2. Case Conversion Functions

a) UPPER () - Convert to uppercase:

SQL>select sid, upper(sname) as sname from student;

SID	SNAME
101	NISANTH
102	SIVAKUMAR
103	SACHIN
104	SIVA

b) LOWER () - Convert to lowercase:

SQL> select sid, lower(dpmt) as dpmt from student;

SID	DPMT
101	mca
102	mca
103	mca
104	mca

c) INITCAP () - Convert first letter of each word to uppercase:

SQL> select sid, initcap(dpmt) as dpmt from student;

SID	DPMT
101	Mca
102	Mca
103	Mca
104	Mca

3. Character Functions

a) SUBSTR () - Extract substring:

SQL> select sid, substr(email, 1, 10) as email from student;

SID	EMAIL
101	nisanthg.2
102	sivakumar
103	sachins.24
104	sivak.24mc

b) LENGTH() - Get the length of a string:

SQL>select sid, length(sname) as sname from student;

SID	SNAME
101	7
102	9
103	6
104	4

4. Date Functions

1. SYSDATE: Returns the current date and time.

SQL>select sysdate as current_date from dual;

CURRENT_DATE

2024-10-06 12:34:56

2. TO_DATE: Converts a string to a date.

SQL>select to_date('2024-10-06', 'yyyy-mm-dd') as converted_date from dual;

CONVERTED_DATE

2024-10-06 00:00:00

3. TO_CHAR: **Converts a date to a string in a specified format**

SQL> select to_char(sysdate, 'dd-mon-yyyy') as formatted_date from dual;

FORMATTED_DATE

06-OCT-20254

5. Number Functions

1. MOD: Returns the remainder of a division operation.

SQL>SELECT MOD(10, 3) as modulus_value FROM dual;

MODULUS_VALUE

1

2. POWER: Returns a number raised to the power of another number.

SQL>SELECT POWER(2, 3) AS power_value FROM dual;

POWER_VALUE

8

3. SQRT: Returns the square root of a number.

SQL>SELECT SQRT(16) AS square_root FROM dual;

SQUARE_ROOT

4

4. ROUND: Rounds a number to a specified number of decimal places.

SQL>SELECT ROUND(123.4567, 2) AS rounded_value FROM dual;

ROUNDED_VALUE

123.46

AGGREGATE FUNCTIONS:

1. Average
2. Count
3. Max
4. Min
5. Sum

Table creation:

create table student (

sid int(10),

```
    sname varchar(20),
    dpmt varchar(5),
    email varchar(25),
    contactno numeric(10)
);
```

Table created.

SQL> **insert into student values**

(101, 'Nisanth', 'MCA', 'nisanthg.24mca@kongu.edu', 9345292781),

(102, 'Sivakumar', 'MCA', 'sivakumarp.24mca@kongu.edu', 8072363074),

(103, 'Sachin', 'MCA', 'sachins.24mca@kongu.edu', 8754681258),

(104, 'Siva', 'MCA', 'sivak.24mca@kongu.edu', 6380603146);

SQL> select * from student;

SID	SNAME	DPMT	EMAIL	CONTACTNO
101	Nisanth	MCA	nisanthg.24mca@kongu.edu	9345292781
102	Sivakumar	MCA	sivakumarp.24mca@kongu.edu	8072363074
103	Sachin	MCA	sachins.24mca@kongu.edu	8754681258
104	Siva	MCA	sivak.24mca@kongu.edu	6380603146

1. Minimum (MIN): Find the minimum contactno.

SQL> SELECT MIN (contactno) FROM student;

MIN(CONTACTNO)

6380603146

2. Maximum (MAX): Find the maximum contactno**.**

SQL> SELECT MAX (contactno) FROM student;

MAX(CONTACTNO)

9345292781

3. Count (COUNT): Count the total number of students.

SQL> SELECT COUNT (sid) FROM student;

COUNT(SID)

4

4. Sum (SUM): Calculate the total sum of SID.

SQL> SELECT SUM (sid) FROM student;

SUM(SID)

410

5. Average (AVG): Calculate the average of SID.

SQL> SELECT AVG (sid) FROM student;

AVG(SID)

102.5

SET OPERATIONS

1. Union
2. Union all
3. Intersect
4. Minus

Table creation:

create table student (

sid int(10),
sname varchar(20),
dpmt varchar(5),
email varchar(25),
contactno numeric(10)
);

Table created.

SQL> **insert into student values**

(101, 'Nisanth', 'MCA', 'nisanthg.24mca@kongu.edu', 9345292781),

(102, 'Sivakumar', 'MCA', 'sivakumarp.24mca@kongu.edu', 8072363074),

(103, 'Sachin', 'MCA', 'sachins.24mca@kongu.edu', 8754681258),

(104, 'Siva', 'MCA', 'sivak.24mca@kongu.edu', 6380603146);

SQL> select * from student;

SID	SNAME	DPMT	EMAIL	CONTACTNO
101	Nisanth	MCA	nisanthg.24mca@kongu.edu	9345292781
102	Sivakumar	MCA	sivakumarp.24mca@kongu.edu	8072363074
103	Sachin	MCA	sachins.24mca@kongu.edu	8754681258
104	Siva	MCA	sivak.24mca@kongu.edu	6380603146

create table course (

course_id int(10) primary key,
course_name varchar(50) not null,
department varchar(30),
credits int(3));

Table created.

insert into course values

(1, 'DBMS', 'CS', 3),

(2, 'ML', 'CS', 4),

(3, 'DM', 'Business', 3),

(4, 'DSA', 'CS', 4),

(5, 'ML', 'CS', 4),

(6, 'DSA', 'CS', 4),

COURSE_ID	COURSE_NAME	DEPARTMENT	CREDITS
1	DBMS	CS	3
2	ML	CS	4
3	DM	BUSINESS	3
4	DSA	CS	4
5	ML	CS	4
6	DSA	CS	4

1. UNION

SQL>select sname as name from student
union
select course_name as name from course;

NAME
Nisanth
Sivakumar
Sachin
Siva

DBMS
ML
DM
DSA

2. UNION ALL

SQL>select sname as name from student
union all
select course_name as name from course;

NAME
Nisanth
Sivakumar
Sachin
Siva
DBMS
ML
DM
DSA
ML
DSA

3. INTERSECT

SQL> select sname as name from student
intersect
select course_name as name from course;

NAME

(empty, as there are no common names)

4. MINUS

SQL>select sname as name from student
minus
select course_name as name from course;

NAME
Nisanth
Sivakumar
Sachin
Siva

COE (30)	
RECORD (20)	
VIVA (10)	
TOTAL (60)	

RESULT:

Thus, the execution of single row functions, aggregate functions and set operations for University Database system has been done successfully.