

TASK 3 – DATATYPE_CONSTRAINTS (06/07/23)

Question1:

Write a small explanation of all commands in Data Control Language (DCL) and Transaction Control Language (TCL), along with some examples.

Answer

Data Control Language (DCL) –

GRANT: This command gives users access privileges to the database.

Syntax:

GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;

Eg: Grant select, Insert on Users To 'user2'@localhost

REVOKE: This command withdraws the user's access privileges given by using the GRANT command.

Syntax:

REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;

Eg: Revoke select, Insert on Users from 'user2'@localhost

Transaction Control Language (TCL)-

COMMIT: Commit a transaction

Syntax: Commit

Eg: Update products set product_name = "Sample" where product_id=1

Commit;

ROLLBACK & SAVEPOINT:

Rollbacks a transaction in case of any error occurs & Sets a save point within a transaction.

Syntax: Rollback;

Savepoint S1;

Eg: Begin

Update

SAVEPOINT S1;

Delete

SAVEPOINT S2;

Update

ROLLBACK TO S1;

End

It will roll back to s1 where only update alone will be committed

Question 2:

Create an table "customer" with following columns

Output:

The screenshot shows a database IDE with the following SQL command in the editor:

```
1 create table customer(cust_id SMALLINT,Cust_no MEDIUMINT,cust_Firstname CHAR , cust_Lastname varchar(20),cust_gender varchar(15),
2 cust_dob date , cust_age int(2),cust_revenue decimal , cust_spend float,cust_addr TEXT ,cust_city char,
3 cust_comments TEXT , cust_status ENUM('ACTIVE','INACTIVE'),cust_available boolean)
4
5 desc customer
```

Below the editor, the 'Result Grid' displays the table schema:

Field	Type	Null	Key	Default	Extra
cust_id	smallint	YES		NULL	
Cust_no	mediumint	YES		NULL	
cust_Firstname	char(30)	YES		NULL	
cust_Lastname	varchar(20)	YES		NULL	
cust_gender	varchar(15)	YES		NULL	
cust_dob	date	YES		NULL	
cust_age	int	YES		NULL	
cust_revenue	decimal(10,0)	YES		NULL	
cust_spend	float	YES		NULL	
cust_addr	text	YES		NULL	
cust_city	char(1)	YES		NULL	
cust_comments	text	YES		NULL	
cust_status	enum('ACTIV...')	YES		NULL	
cust_available	tinyint(1)	YES		NULL	

Query:

```
create table customer(cust_id SMALLINT,Cust_no MEDIUMINT,cust_Firstname CHAR ,
cust_Lastname varchar(20),cust_gender varchar(15),cust_dob date , cust_age
int(2),cust_revenue decimal , cust_spend float,cust_addr TEXT ,cust_city
char,cust_comments TEXT , cust_status ENUM('ACTIVE','INACTIVE'),cust_available boolean)
```

Question 3:

The output of the SELECT statement should be a single column

Output:

The screenshot shows a database IDE with the following SQL query in the editor:

```
790
791 SELECT cust_id, CAST(cust_id AS CHAR) AS cust_id_string
792 FROM customer;
793
794
```

Below the editor, the 'Result Grid' displays the query results:

cust_id	cust_id_string
1	1
2	2
3	3
4	4

i.

ii.

```
792 FROM customer;  
793 • SELECT cust_id, STR_TO_DATE(cust_dob, '%Y-%m-%d') AS cust_dob_date  
794 FROM customer;
```

	cust_id	cust_dob_date
▶	1	1999-04-04
	2	1998-04-07
	3	1999-04-07
	4	1999-08-07

iii.

```
794 FROM customer;  
795 • SELECT cust_id, CAST(cust_revenue AS SIGNED) AS cust_revenue_int  
796 FROM customer;
```

	cust_id	cust_revenue_int
▶	1	27000
	2	27000
	3	27000
	4	27000

iv.

```
797 • SELECT cust_id, CAST(cust_available AS CHAR) AS cust_available_text  
798 FROM customer;
```

	cust_id	cust_available_text
▶	1	1
	2	1
	3	1
	4	1

Query:

```
SELECT CAST(cust_id AS CHAR) AS cust_id_string
```

```
FROM customer;
```

```
SELECT CAST(cust_dob as Date) AS cust_dob_date
```

```
FROM customer;
```

```
SELECT CAST(cust_revenue AS SIGNED) AS cust_revenue_int
```

```
FROM customer;
```

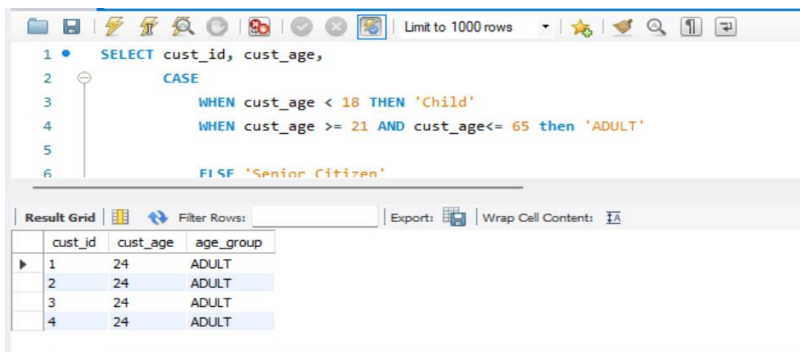
```
SELECT CAST(cust_available AS CHAR) AS cust_available_text
```

```
FROM customer;
```

Question 4:

Case statement

Output:



The screenshot shows a SQL IDE with a query editor and a results grid. The query is a SELECT statement with a CASE statement to categorize customers by age. The results grid shows four rows, all categorized as 'ADULT'.

```
1 • SELECT cust_id, cust_age,  
2     CASE  
3         WHEN cust_age < 18 THEN 'Child'  
4         WHEN cust_age >= 21 AND cust_age <= 65 THEN 'ADULT'  
5     ELSE 'Senior Citizen'  
6
```

	cust_id	cust_age	age_group
1	1	24	ADULT
2	2	24	ADULT
3	3	24	ADULT
4	4	24	ADULT

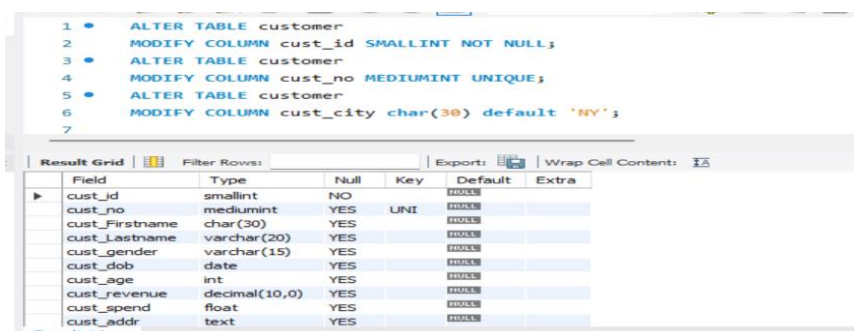
Query:

```
SELECT cust_id, cust_age,  
  
CASE  
  
    WHEN cust_age < 18 THEN 'Child'  
  
    WHEN cust_age >= 21 AND cust_age <= 65 THEN 'ADULT'  
  
ELSE 'Senior Citizen'  
  
END AS age_group  
  
FROM customer;
```

Question 5:

Alter the above table by adding the following constraints

Output:



The screenshot shows a SQL IDE with an ALTER TABLE query and a table structure grid. The query adds constraints to the 'customer' table: a SMALLINT NOT NULL constraint on cust_id, a MEDIUMINT UNIQUE constraint on cust_no, and a CHAR(30) default 'NY' constraint on cust_city. The table structure grid shows the resulting table with columns: cust_id (smallint, NOT NULL, PRIMARY KEY), cust_no (mediumint, YES, UNIQUE), cust_firstname (char(30), YES), cust_lastname (varchar(20), YES), cust_gender (varchar(15), YES), cust_dob (date, YES), cust_age (int, YES), cust_revenue (decimal(10,0), YES), cust_spend (float, YES), and cust_addr (text, YES).

```
1 • ALTER TABLE customer  
2     MODIFY COLUMN cust_id SMALLINT NOT NULL;  
3 • ALTER TABLE customer  
4     MODIFY COLUMN cust_no MEDIUMINT UNIQUE;  
5 • ALTER TABLE customer  
6     MODIFY COLUMN cust_city char(30) default 'NY';  
7
```

Field	Type	Null	Key	Default	Extra
cust_id	smallint	NO			PRIMARY
cust_no	mediumint	YES	UNI		
cust_firstname	char(30)	YES			
cust_lastname	varchar(20)	YES			
cust_gender	varchar(15)	YES			
cust_dob	date	YES			
cust_age	int	YES			
cust_revenue	decimal(10,0)	YES			
cust_spend	float	YES			
cust_addr	text	YES			

Query

ALTER TABLE customer

MODIFY COLUMN cust_id SMALLINT NOT NULL;

ALTER TABLE customer

MODIFY COLUMN cust_no MEDIUMINT UNIQUE;

ALTER TABLE customer

MODIFY COLUMN cust_city char(30) default 'NY';

Question 6 :

Create a table called 'Employees' with the following columns:

Output:

	Field	Type	Null	Key	Default	Extra
▶	emp_id	int	NO	PRI	NULL	
	emp_name	varchar(50)	NO		NULL	
	emp_age	int	YES		NULL	
	emp_dept	varchar(50)	YES		IT	

Query:

create table employees (emp_id int primary key,emp_name varchar(50) not null,
emp_age int,emp_dept varchar(50) default 'IT' , CHECK(emp_age >= 18))

Question 7:

Create an table "customers_data" with below columns

Output:

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content:
	Field	Type	Null	Key	Default	Extra
▶	cust_id	int	NO	PRI	NULL	
	cust_name	varchar(30)	YES		NULL	

	Field	Type	Null	Key	Default	Extra
▶	order_id	int	NO	PRI	NULL	
	cust_id	int	NO	MUL	NULL	
	order_date	date	YES		NULL	

Query:

```
create table customer_data (cust_id int primary key,cust_name varchar(30));
```

```
INSERT INTO customer_data(cust_id,cust_name) values(1,'Vishal');
```

```
INSERT INTO customer_data(cust_id,cust_name) values(2,'Aiyam');
```

```
INSERT INTO customer_data(cust_id,cust_name) values(3,'Perumal');
```

```
create table orders (order_id int primary key,cust_id int not null , order_date date,
```

```
FOREIGN KEY (cust_id) references customer_data(cust_id) )
```

```
insert into orders(order_id,
```

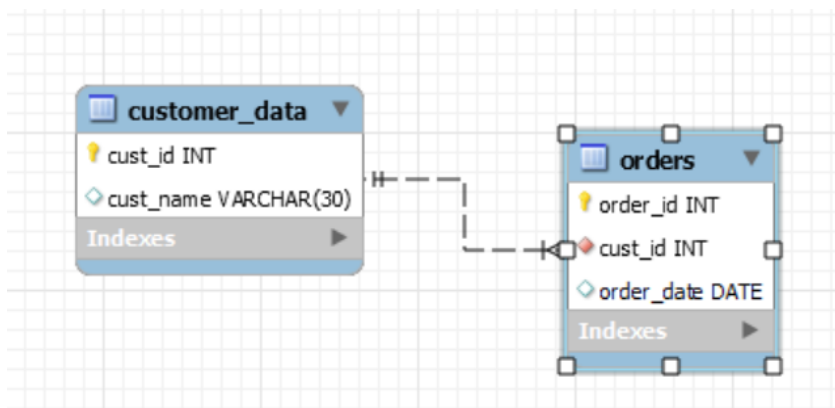
```
cust_id,
```

```
order_date) values (1,2,'1999-04-04');
```

```
insert into orders(order_id,
```

```
cust_id,
```

```
order_date) values (2,2,'1999-04-04');
```



Steps:

1. Created **customer_data** table and inserted data into it
2. Created **orders** table with constraint like **FOREIGN KEY REFERENCES TO (Primary key of other table)** i.e – **Cust_id**

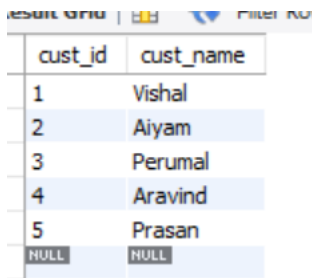
3. Insertion of data into orders table should match with values of cust_id of customer table, if not error will be thrown

4. In order of deletion, first data in order table need to be deleted before deleting in Customer table, if not error will be thrown

Question 8

Enter some random 5 records of data into customers_data and orders table.

Output:



cust_id	cust_name
1	Vishal
2	Aiyam
3	Perumal
4	Aravind
5	Prasan
NULL	NULL

Query:

```
INSERT INTO customer_data(cust_id,cust_name) values(1,'Vishal');
```

```
INSERT INTO customer_data(cust_id,cust_name) values(2,'Aiyam');
```

```
INSERT INTO customer_data(cust_id,cust_name) values(3,'Perumal');
```

```
INSERT INTO customer_data(cust_id,cust_name) values(4,'Aravind');
```

```
INSERT INTO customer_data(cust_id,cust_name) values(5,'Prasan');
```

```
insert into orders(order_id,cust_id,order_date) values (1,1,'1999-04-04');
```

```
insert into orders(order_id,cust_id,order_date) values (2,2,'2000-02-25');
```

```
insert into orders(order_id,cust_id,order_date) values (3,3,'1999-12-25');
```

```
insert into orders(order_id,cust_id,order_date) values (4,5,'1999-12-11');
```

Question 9:

Remove any of 2 records from the customers data and orders table.

Steps:

1. Remove data from Orders table first where in this case I deleted cust_id = 3 and 5 in order table
2. Now you can able to remove cust_id 3 and 5 from Customer_data table

Query:

delete from orders where cust_id in (3,5)

delete from customer_data where cust_id in (3,5)

Question 10:

Difference between unique and primary key

Difference between float, double and decimal

Answers:

PRIMARY	UNIQUE
Primary key does not store null values	Unique key stores null values
Table can contain only one primary key	Table can contain multiple unique keys
We can't able to delete or modify values of primary key	We can able to change values of unique key
The primary Key is used for indicating the rows uniquely.	The Unique Key is used for preventing duplicate entries.
Used to serve as a unique identifier for each row in a table	Uniquely determines a row that isn't the primary key.

FLOAT	DOUBLE	DECIMAL
M should be greater than D Eg: Float(4,2)	M should be greater than D Eg: Double(4,2)	Convert to Whole , if no precision
32 bits	64 bits	128bits
Values are less accurate	More accurate than float	More accurate than float and double
Upto 24 precision	Greater than 24 precision	

