IRC_RDBMS_D3_DDL_CE_COD

Test Summary

- No. of Sections: 1
- No. of Questions: 5
- Total Duration: 50 min

Section 1 - Commands

Section Summary

- No. of Questions: 5
- Duration: 50 min

Additional Instructions:
None

Q1. Write a query to create a table **employees** .as per below given details.

Note:
1. Make sure to apply **NOT NULL** and Primary Key (PK) condition for the columns where ever applicable.
2. While creating a table, table name and column name should be exactly same as given in DDL (including upper/lower case)

Table name: employees

Column Name	Data Type	Can be Nullable?
employee_id (PK)	int	No
first_name	varchar(25)	No
last_name	varchar(25)	No
email	varchar(25)	No
phone_number	varchar(20)	No
hire_date	date	No
job_id	varchar(10)	Yes
salary	decimal(8,2)	No
commission_pct	int	Yes
manager_id	int	Yes
department_id	int	Yes

Sample Input

Sample Output

```
Field   Type    Null    Key     Default Extra
employee_id     int(11) NO      PRI     NULL
first_name      varchar(25)     NO              NULL
```

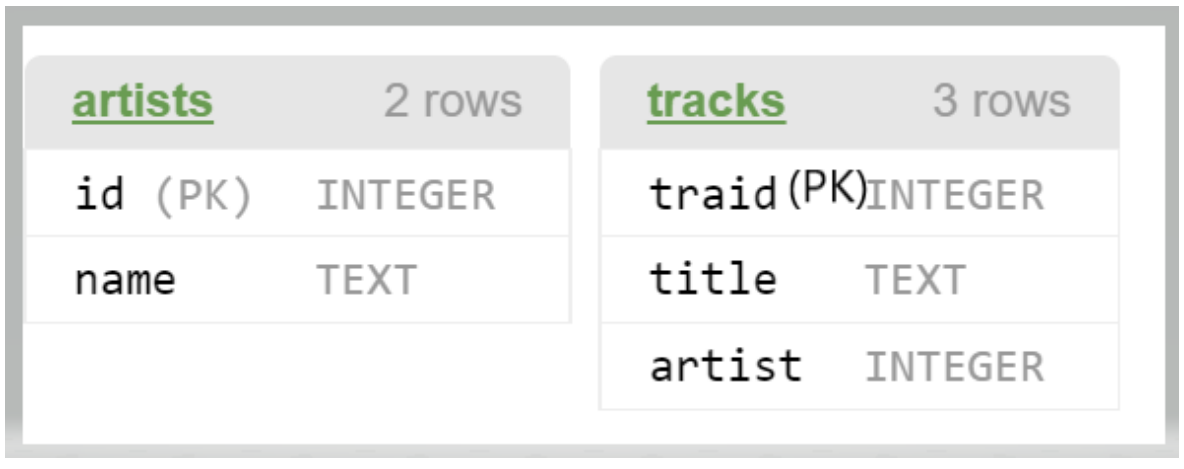
Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. From the table schema given below, write a query to create a table **tracks**.
(Follow the same table names/column names as given along with the case)

The **artists** table is already created with primary key as **id.**

Note:
In **tracks** table, **traid** is the Primary key and **artist** is the Foreign Key referencing id of artists table.

While creating tracks table,
- traid is the Primary Key
- define the foreign key relation between 2 tables.
- define all the columns with NOT NULL condition



Input Format

The artists table is created in the backend.

Sample Input

Sample Output

```
TABLE_NAME      COLUMN_NAME     CONSTRAINT_NAME
tracks  traid   PRIMARY NULL    NULL
tracks  artist  tracks_ibfk_1   artists id
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. From the DDL given below,

Write a query to create a table "**facilities**"

Hint: Refer the table for the field names and its data type of the table.

Table name: facilities

Column Name	Data Type	Can be Nullable?
facid(PK)	int	No
name	varchar(100)	No
membercost	int	No
guestcost	int	No
initialoutlay	int	Yes
monthlymaintenance	int	Yes

Note:

1. Make sure to apply NOT NULL and Primary Key (PK) condition for the columns where ever applicable.
2. While creating a table, table name and column name should be exactly same as given in DDL (including upper/lower case)

Sample Input

Sample Output

```
Field   Type    Null    Key     Default Extra
facid   int(11) NO      PRI     NULL
name    varchar(100)    NO              NULL
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. Write a query to change the column name **zipcode** to **pincode** in members table.

Table name: members

Column Name	Data Type	Can be Nullable?
memid(PK)	int	No
surname	varchar(200)	No
firstname	varchar(200)	No
address	varchar(300)	No
zipcode	int	No
telephone	varchar(20)	No
recommendedby	int	Yes
joindate	varchar(10)	No

Input Format

The members table is already created.

Sample Input

Sample Output

```
COLUMN_NAME
memid
surname
firstname
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. Write a query to drop the table **facilities**

Input Format

The required tables are populated in the backend.

Sample Input

Sample Output

```
Tables_in_ri_db
bookings
members
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - Commands

Q1

Test Case

Input

Output

```
Field   Type    Null    Key     Default Extra
employee_id     int(11) NO      PRI     NULL
first_name      varchar(25)     NO              NULL
```

Weightage - 100

Sample Input

Sample Output

```
Field   Type    Null    Key     Default Extra
employee_id     int(11) NO      PRI     NULL
first_name      varchar(25)     NO              NULL
```

Solution

```
CREATE TABLE employees (
    employee_id int PRIMARY KEY NOT NULL,
    first_name VARCHAR (25) NOT NULL,
    last_name VARCHAR (25) NOT NULL ,
    email VARCHAR (25) NOT NULL ,
    phone_number VARCHAR (20) NOT NULL ,
    hire_date DATE NOT NULL,
    job_id VARCHAR (10),
    salary DECIMAL (8, 2) NOT NULL ,
    commission_pct INT ,
    manager_id INT ,
    department_id INT
);
```

Footer

```
show columns from employees;
```

Q2

Test Case

Input

Output

```
TABLE_NAME      COLUMN_NAME     CONSTRAINT_NAME
tracks  traid   PRIMARY NULL    NULL
tracks  artist  tracks_ibfk_1   artists id
```

Weightage - 100

Sample Input

Sample Output

```
TABLE_NAME      COLUMN_NAME     CONSTRAINT_NAME
tracks  traid   PRIMARY NULL    NULL
tracks  artist  tracks_ibfk_1   artists id
```

Solution

```
CREATE TABLE tracks (
  traid  integer primary key NOT NULL,
  title   TEXT NOT NULL,
  artist INTEGER NOT NULL,
  FOREIGN KEY(artist) REFERENCES artists(id)
);
```

Footer

```
SELECT
  TABLE_NAME,COLUMN_NAME,CONSTRAINT_NAME, REFERENCED_TABLE_NAME,REFERENCED_COLUMN_NAME
FROM
  INFORMATION_SCHEMA.KEY_COLUMN_USAGE
WHERE TABLE_NAME = 'tracks';
```

Q3

Test Case

Input

Output

```
Field   Type    Null    Key     Default Extra
facid   int(11) NO      PRI     NULL
name    varchar(100)    NO              NULL
```

Weightage - 100

Sample Input

Sample Output

```
Field   Type    Null    Key     Default Extra
facid   int(11) NO      PRI     NULL
name    varchar(100)    NO              NULL
```

Solution

```
CREATE TABLE facilities (
    facid int primary key NOT NULL ,
    name varchar(100) NOT NULL,
    membercost int NOT NULL,
    guestcost int NOT NULL,
    initialoutlay int,
    monthlymaintenance int);
```

Footer

```
SHOW COLUMNS FROM facilities;
```

Q4

Test Case

Input

Output

```
COLUMN_NAME
memid
surname
firstname
```

Weightage - 100

Sample Input

Sample Output

```
COLUMN_NAME
memid
surname
firstname
```

Solution

```
ALTER TABLE members
CHANGE zipcode pincode int;
```

Footer

```
SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE
TABLE_NAME = 'members';
```

Q5

Test Case

Input

Output

```
Tables_in_ri_db
bookings
members
```

Weightage - 100

Sample Input

Sample Output

```
Tables_in_ri_db
bookings
members
```

Solution

```
DROP TABLE facilities;
```

Footer

```
SHOW TABLES;
```