

CSE 4308
Database Management Systems Lab

Lab-04

Group-B

Mohammad Mahbub Ur Rahman
220042148
BSc. in Software Engineering

Tasks

1.

```
-- 1
CREATE TABLE pokemon(
    pokemon_id INT NOT NULL,
    name VARCHAR(50),
    type VARCHAR(20),
    hp INT,
    attack INT,
    defence INT,
    speed INT,
    CONSTRAINT PK_POKEMON_ID PRIMARY KEY (pokemon_id)
```

Here I am creating pokemon table. And setting pokemon_id as primary key.

```
CREATE TABLE trainer(
    trainer_id INT NOT NULL,
    first_name VARCHAR(30),
    last_name VARCHAR(30),
    city VARCHAR(30),
    CONSTRAINT PK_TRAINER_ID PRIMARY KEY (trainer_id)
);
```

Here I am creating trainer table. And setting trainer_id as primary key.

```
INSERT INTO pokemon (pokemon_id, name, type, hp, attack, defence, speed) VALUES
(1, "Bulbasaur", "Grass", 45, 49, 49, 45),
(2, "Ivysaur", "Grass", 60, 62, 63, 60),
(3, "Venusaur", "Grass", 80, 82, 83, 80),
(4, "Charmander", "Fire", 39, 52, 43, 65),
(5, "Charmeleon", "Fire", 58, 64, 58, 80),
(6, "Charizard", "Fire", 78, 84, 78, 100),
(7, "Squirtle", "Water", 44, 48, 65, 43),
(8, "Wartortle", "Water", 59, 63, 80, 58),
(9, "Blastoise", "Water", 79, 83, 100, 78),
(10, "Pikachu", "Electric", 35, 55, 40, 90),
(11, "Raichu", "Electric", 60, 90, 55, 110);
```

Inserting data into pokemon table.

```
INSERT INTO trainer (trainer_id, first_name, last_name, city) VALUES
(1, "Ash", "Ketchum", "Pallet Town"),
(2, "Misty", "Williams", "Cerulean City"),
(3, "Brock", "Harrison", "Pewter City"),
(4, "Gary", "Oak", "Pallet Town"),
(5, "Erika", "Green", "Celadon City");
```

Inserting data into trainer table.

2.

```
-- 2
SELECT DISTINCT type
FROM pokemon;
```

Here the distinct keyword selects only the unique ones and ignores the duplicate data.

Output:

type
Grass
Fire
Water
Electric

3.

```
SELECT *  
FROM pokemon  
WHERE attack BETWEEN 50 AND 80;
```

Here the between keyword only selects the data of attack if it is (50 <= attack <= 80)

Output:

pokemon_id	name	type	hp	attack	defence	speed
2	Ivysaur	Grass	60	62	63	60
4	Charmander	Fire	39	52	43	65
5	Charmeleon	Fire	58	64	58	80
8	Wartortle	Water	59	63	80	58
10	Pikachu	Electric	35	55	40	90

4

```
-- 4  
SELECT *  
FROM pokemon  
WHERE name LIKE 'C%';
```

Here like keyword compares the string given after it with the name and gives us the matching ones. The % in string means substrings. 'C%' means the name starts with C and then it can have any number of characters.

Output:

pokemon_id	name	type	hp	attack	defence	speed
4	Charmander	Fire	39	52	43	65
5	Charmeleon	Fire	58	64	58	80
6	Charizard	Fire	78	84	78	100

5

```
-- 5
SELECT *
FROM pokemon
WHERE name LIKE '%saur%';
```

Same as the last one. Here '%saur%' means that the name should contain 'saur' and any number of character/substrings can come before or after it.

Output:

pokemon_id	name	type	hp	attack	defence	speed
1	Bulbasaur	Grass	45	49	49	45
2	Ivysaur	Grass	60	62	63	60
3	Venusaur	Grass	80	82	83	80

6

```
-- 6
SELECT *
FROM pokemon
WHERE name LIKE '____e____';
```

Here '_' specifies the character count in name and 'e' sets that the 5th character in name must be e

Output:

pokemon_id	name	type	hp	attack	defence	speed
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Null cause no such name exists.

7

```
-- 7
SELECT concat(first_name, " ", last_name) as name, city
FROM trainer;
```

Here concat() takes multiple strings as input and returns a single string containing those input.

Output:

name	city
Ash Ketchum	Pallet Town
Misty Williams	Cerulean City
Brock Harrison	Pewter City
Gary Oak	Pallet Town
Erika Green	Celadon City

8

```
-- 8
SELECT *
FROM pokemon
ORDER BY type ASC, attack DESC;
```

Here 'order by' is used to order the result. 'Type ASC' means the data will be ordered first by 'type' in ascending order then if there is more than one data of 'type' then those data will be sorted in descending order.

Output:

pokemon_id	name	type	hp	attack	defence	speed
11	Raichu	Electric	60	90	55	110
10	Pikachu	Electric	35	55	40	90
6	Charizard	Fire	78	84	78	100
5	Charmeleon	Fire	58	64	58	80
4	Charmander	Fire	39	52	43	65
3	Venusaur	Grass	80	82	83	80
2	Ivysaur	Grass	60	62	63	60
1	Bulbasaur	Grass	45	49	49	45
9	Blastoise	Water	79	83	100	78
8	Wartortle	Water	59	63	80	58
7	Squirtle	Water	44	48	65	43

9

```
CREATE TABLE trainer_pokemon(  
    trainer_id INT NOT NULL,  
    pokemon_id INT NOT NULL  
);
```

Created trainer_pokemon table.

10

```
ALTER TABLE trainer_pokemon  
ADD CONSTRAINT FK_triner_id FOREIGN KEY (trainer_id) REFERENCES trainer(trainer_id)  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

Here I am adding foreign key constraint in trainer_pokemon table referencing trainer tables trainer_id. Cascade keyword updates the primary key data if the foreign key data is changed and vice versa.

```
ALTER TABLE trainer_pokemon  
ADD CONSTRAINT FK_pokemon_id FOREIGN KEY (pokemon_id) REFERENCES pokemon(pokemon_id)  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

Same as before