

--List the 8 Kanto gym leaders

```
SELECT distinct t.First_Name
FROM TRAINER t
    INNER JOIN GYM_LEADER gl ON t.Trainer_ID = gl.Trainer_ID
    INNER JOIN GAME_TRAINERS gt ON gl.Trainer_ID = gt.TRAINER_Trainer_ID
    INNER JOIN GAME g ON gt.GAME_Game_ID = g.GAME_ID
    INNER JOIN GENERATION gen ON g.GENERATION_Gen_ID = gen.Gen_ID
    INNER JOIN REGION r ON gen.Gen_ID = r.GENERATION_Gen_ID
WHERE
    Region_Name = 'Kanto';
```

--What Pokemon Type does Giovanni use?

```
SELECT t.First_Name, pt.Type_Name
FROM Trainer t
    INNER JOIN POKEMON_TYPE pt ON t.POKEMON_TYPE_Type_ID = pt.TYPE_ID
WHERE
    First_Name = 'Giovanni';
```

--What games are Maylene in?

```
SELECT t.First_Name, g.Game_Name
FROM TRAINER t
    INNER JOIN GAME_TRAINERS gt ON t.Trainer_ID = gt.TRAINER_Trainer_ID
    INNER JOIN GAME g ON gt.GAME_Game_ID = g.Game_ID
WHERE
    First_Name = 'Maylene';
```

--List all the trainers that use Bug type or Water type

```
SELECT t.First_Name, pt.Type_Name
FROM Trainer t
    INNER JOIN POKEMON_TYPE pt ON t.POKEMON_TYPE_Type_ID = pt.TYPE_ID
WHERE
    Type_Name IN('Bug', 'Water');
```

--Who are all the trainers with last names?

```
SELECT *
FROM TRAINER
WHERE Last_Name IS NOT NULL;
```

--Any name starting with F

```
SELECT *
```

```
FROM TRAINER
WHERE First_Name LIKE 'F%';
```

--How many gym leaders are in Gen 4?

```
SELECT count(distinct gl.Trainer_ID)
FROM TRAINER t
  INNER JOIN GYM_LEADER gl ON t.Trainer_ID = gl.Trainer_ID
  INNER JOIN GAME_TRAINERS gt ON gl.Trainer_ID = gt.TRAINER_Trainer_ID
  INNER JOIN GAME g ON gt.GAME_Game_ID = g.Game_ID
  INNER JOIN GENERATION gen ON g.GENERATION_Gen_ID = gen.Gen_ID
WHERE Gen_Number='IV';
```

--Who are the Gym Leaders of Gen VI and what are Pokemon Type do they possess?

```
SELECT distinct t.First_Name, pt.Type_Name
FROM TRAINER t
  INNER JOIN POKEMON_TYPE pt ON t.POKEMON_TYPE_Type_ID = pt.Type_ID
  INNER JOIN GYM_LEADER gl ON t.Trainer_ID = gl.Trainer_ID
  INNER JOIN GAME_TRAINERS gt ON gl.Trainer_ID = gt.TRAINER_Trainer_ID
  INNER JOIN GAME g ON gt.GAME_Game_ID = g.Game_ID
  INNER JOIN GENERATION gen ON g.GENERATION_Gen_ID = gen.Gen_ID
WHERE Gen_Number='VI';
```

--Return the Pokemon Types from most used to least used

```
WITH typeCount AS (
  SELECT t.POKEMON_TYPE_Type_ID, COUNT(t.POKEMON_TYPE_Type_Id) AS CountedTypes
  FROM TRAINER t
  GROUP BY t.POKEMON_TYPE_Type_ID)
SELECT *
FROM (
  SELECT pt.Type_Name, typeCount.CountedTypes
  FROM POKEMON_TYPE pt
  INNER JOIN typeCount ON pt.Type_ID =
typeCount.POKEMON_TYPE_Type_ID
  ORDER BY typeCount.CountedTypes DESC);
--WHERE ROWNUM = 1;
```

--What Pokemon types have the highest frequency?

```
WITH typeCount AS (
  SELECT t.POKEMON_TYPE_Type_ID, COUNT(t.POKEMON_TYPE_Type_Id) AS CountedTypes
  FROM TRAINER t
```

```

        GROUP BY t.POKEMON_TYPE_Type_ID)
, maxTypeCount AS (    SELECT t.POKEMON_TYPE_Type_ID, COUNT(t.POKEMON_TYPE_Type_Id) AS
CountedTypes
        FROM TRAINER t
        GROUP BY t.POKEMON_TYPE_Type_ID)
SELECT *
FROM (
        SELECT pt.Type_Name
        FROM POKEMON_TYPE pt
                INNER JOIN typeCount ON pt.Type_ID =
typeCount.POKEMON_TYPE_Type_ID
                INNER JOIN maxTypeCount ON pt.Type_ID =
maxTypeCount.POKEMON_TYPE_Type_ID
        WHERE typeCount.CountedTypes = (SELECT MAX(maxTypeCount.CountedTypes) FROM
maxTypeCount)
        ORDER BY typeCount.CountedTypes DESC);

```

--What are the Town Names in Generation II?

```

SELECT distinct t.TOWN_NAME
FROM TOWN t
        INNER JOIN GAME_TOWNS gt ON t.Town_ID = gt.TOWN_Town_ID
        INNER JOIN GAME g ON gt.GAME_Game_ID = g.GAME_ID
        INNER JOIN GENERATION gen ON g.GENERATION_Gen_ID = gen.Gen_ID

WHERE
Gen_Number = 'II';

```

--List all the trainers with a town, if they have one

```

SELECT *
FROM Trainer t
LEFT OUTER JOIN TOWN tow ON t.Trainer_ID = tow.GYM_LEADER_TRAINER_ID;

```