



----- Part 2 -----

```
/*
----- Foreign Key in Detail -----

*/
```

```
USE college;
CREATE TABLE dept (
    id INT PRIMARY KEY,
    subject_name VARCHAR(50)
);
```

```
CREATE TABLE teacher (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES dept(id)
);
```

```
-- We can see the connection between these two table in the er diagram provided in github

-- dept -> parent table
-- teacher -> child table
```

----- Cascading for Foreign Key -----

```
/*
On Delete Cascade:
-----
When we create a foreign key using this option, it deletes the referencing rows in the child table
when the referenced row is deleted in the parent table which has a primary key.
```

```
On Update Cascade:
-----
When we create a foreign key using UPDATE CASCADE the referencing rows are updated in the child
table when the referenced row is updated in the parent table which has a primary key
*/
```

```
-- so we have to write this create table in this way
CREATE TABLE teacher (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES dept(id)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
```

```
INSERT INTO dept
VALUES
(101, "CSE"),
(102, "IT");
```

```
SELECT * FROM dept;
```

```
INSERT INTO teacher
VALUES
(1001, "jayanta", 101),
(1002, "arindam", 101),
(1003, "sagar", 102),
(1004, "tanaya", 101),
(1005, "yusuf", 102);
```

```
SELECT * FROM teacher;
```

```
UPDATE dept
SET id = 103
WHERE subject_name = "CSE" AND id = 101;
-- this will not just update dept table it will also update teacher table's dept_id
```

----- Table Related Queries -----

```
/*
----- ALTER (to change the schema) -----
-----
```

```
ADD Column
-----
ALTER TABLE table_name
ADD COLUMN column_name datatype constraint;
```

```
DROP Column
-----
ALTER TABLE table_name
DROP COLUMN column_name;
```

```
RENAME Table
-----
ALTER TABLE table_name
RENAME TO new_table_name;
```

```
CHANGE Column (rename):
-----
ALTER TABLE table_name
CHANGE COLUMN old_name new_name new_datatype new_constraint;
```

```
MODIFY Column (modify datatype/constraint):
-----
ALTER TABLE table_name
MODIFY col_name new_datatype new_constraint;
```

```
*/

ALTER TABLE student
ADD COLUMN age INT; -- this will add a new column age in student
```

```
ALTER TABLE student
DROP COLUMN age; -- this will delete the age column from student
```

```
ALTER TABLE student
RENAME TO stu; -- this will rename student table to stu
```

```
ALTER TABLE stu
ADD COLUMN age INT; -- adding new col age
```

```
ALTER TABLE stu
CHANGE age stu_age INT; -- change column name from age to stu_age
```

```
ALTER TABLE stu
MODIFY stu_age VARCHAR(2); -- change column datatype int to varchar
```

```
SELECT * FROM stu;
```

```
/*
----- TRUNCATE -----
-----
```

```
-> DROP TABLE deletes whole table with data whereas
-> TRUNCATE TABLE table_name delete only table's data not the table
*/
```

```
TRUNCATE TABLE stu; -- this will delete all the data but not the table
```

```
/*
----- PRACTICE Qs: -----
```

```
Qs: In the student table:
-> change the name of the column "name" to "full_name".
-> Delete all the students who scored marks less than 80.
-> Delete the column for grades
```

```
*/

SELECT * FROM stu;

--- Ans: Q1
ALTER TABLE stu
CHANGE name full_name VARCHAR(50);

--- Ans: Q2
```

```
DELETE FROM stu
WHERE marks < 80;
```

```
--- Ans: Q3
ALTER TABLE stu
DROP COLUMN grade;
```
