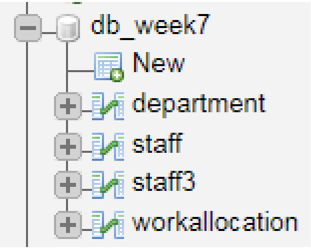
Additional DDL commands: ALTER, DROP & Constraints

DROP

- DROP TABLE
  - Remove a table from the database.
- Example

```
DROP TABLE Staff2;
```

Hints: After successful execution of this statement, refresh your browser and you will see Staff2 does not exists anymore!



db_week7
New
department
staff
staff3
workallocation

ALTER

- ALTER TABLE allows you to change the structure of an existing table.
- For example, you can add or delete columns, {create or destroy indexes}, change the type of existing columns, or rename columns or the table itself.
- **Syntax:**

  ALTER TABLE tbl_name [ADD | MODIFY | DROP],...

ALTER: ADD | MODIFY | DROP column(s)

- **ADD**

  ALTER TABLE Staff

  ADD Address VARCHAR(30);

  - It adds a new column to the table staff

- **MODIFY**

  ALTER TABLE Staff3

  MODIFY Salary INT(11);

  - It changes the data type of the column Phone

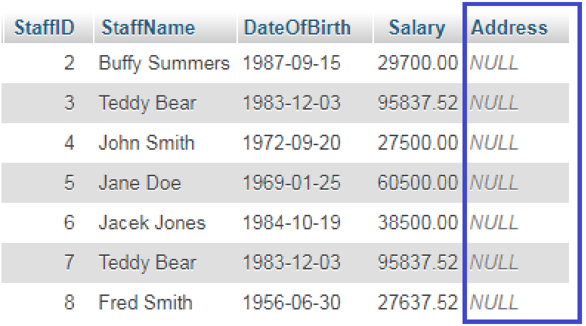
- **DROP**

  ALTER TABLE Staff

  DROP COLUMN Address;

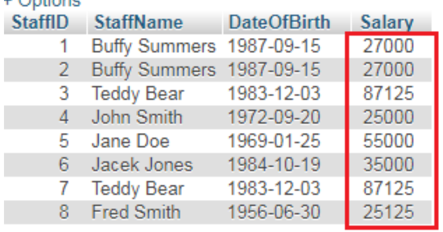
  - It deletes the column 'address' from the table staff

Staff table:



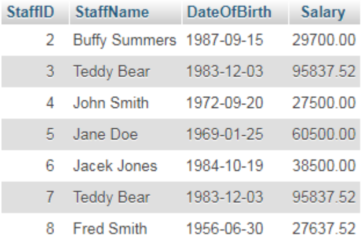
StaffID	StaffName	DateOfBirth	Salary	Address
2	Buffy Summers	1987-09-15	29700.00	NULL
3	Teddy Bear	1983-12-03	95837.52	NULL
4	John Smith	1972-09-20	27500.00	NULL
5	Jane Doe	1969-01-25	60500.00	NULL
6	Jacek Jones	1984-10-19	38500.00	NULL
7	Teddy Bear	1983-12-03	95837.52	NULL
8	Fred Smith	1956-06-30	27637.52	NULL

Staff3 table:



StaffID	StaffName	DateOfBirth	Salary
1	Buffy Summers	1987-09-15	27000
2	Buffy Summers	1987-09-15	27000
3	Teddy Bear	1983-12-03	87125
4	John Smith	1972-09-20	25000
5	Jane Doe	1969-01-25	55000
6	Jacek Jones	1984-10-19	35000
7	Teddy Bear	1983-12-03	87125
8	Fred Smith	1956-06-30	25125

Staff table:



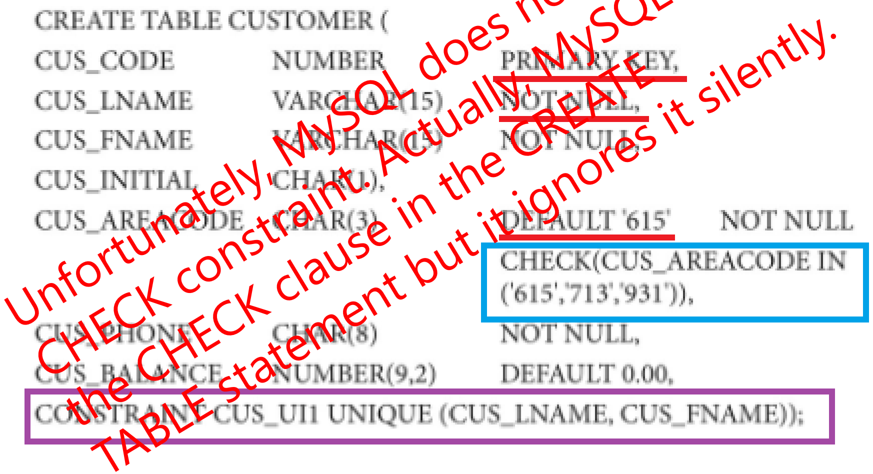
StaffID	StaffName	DateOfBirth	Salary
2	Buffy Summers	1987-09-15	29700.00
3	Teddy Bear	1983-12-03	95837.52
4	John Smith	1972-09-20	27500.00
5	Jane Doe	1969-01-25	60500.00
6	Jacek Jones	1984-10-19	38500.00
7	Teddy Bear	1983-12-03	95837.52
8	Fred Smith	1956-06-30	27637.52

Address column does not exists anymore!

SQL Constraints

- **Entity integrity:** PRIMARY KEY command makes sure the PK is unique and cannot be null.
- **Referential integrity:** FOREIGN KEY command makes sure the FK has a valid value from parent table.
- **NOT NULL** constraint ensures that a column does not accept nulls.
- **UNIQUE** constraint ensures that all values in a column are unique.
- **DEFAULT** constraint assigns a value to an attribute when a new row is added to a table. The end user may, of course, enter a value other than the default value.
- **CHECK** constraint is used to validate data when an attribute value is entered. Also, known as **Legal-Values Integrity** constraint. For example, credit limit check!

```
CREATE TABLE IF NOT EXISTS DEPARTMENT(
        DepartmentID        INT PRIMARY KEY
AUTO_INCREMENT,
        DepartmentName      VARCHAR(30),
        Budget              DOUBLE,
        ManagerID           INT NOT NULL,
        FOREIGN KEY (ManagerID) REFERENCES
Staff(StaffID)
) ENGINE=InnoDB;
```



```
CREATE TABLE CUSTOMER (
CUS_CODE       NUMBER         PRIMARY KEY,
CUS_LNAME      VARCHAR(15)    NOT NULL,
CUS_FNAME      VARCHAR(15)    NOT NULL,
CUS_INITIAL    CHAR(1),
CUS_AREACODE   CHAR(3)        DEFAULT '615'    NOT NULL
                              CHECK(CUS_AREACODE IN
                              ('615','713','931')),
CUS_PHONE      CHAR(8)        NOT NULL,
CUS_BALANCE    NUMBER(9,2)    DEFAULT 0.00,
CONSTRAINT CUS_UI1 UNIQUE (CUS_LNAME, CUS_FNAME));
```

Unfortunately, MySQL does not support CHECK constraint. Actually, MySQL accepts the CHECK clause in the CREATE TABLE statement but it ignores it silently.

Thank you