# DDL Statements

In this lesson, we will learn how to write DDL statements to:

- create new table
- modify table structure
- drop a table

# The general DDL structure for creating a table

- An SQL relation is defined using the **create table** command:

**create table** *r*
(
$A_1$ $D_1$,
$A_2$ $D_2$,
...,
$A_n$ $D_n$,
integrity-constraint$_1$,
...,
integrity-constraint
);

- *r* is the name of the table (e.g. Student)
- each $A_i$ is an attribute name (e.g. Student_ID) in the schema of relation *r*
- $D_i$ is the data type of values in the domain of attribute $A_i$ (e.g. varchar(40))

# Create table syntax

```
Create table table_name
(
      column_name1     datatype(size),
      column_name2     datatype(size),
      column_name3     datatype(size),
      ……
      Integrity_constraints1,
      Integrity_constraints2
);
```

# Create table syntax (con't.)

**Integrity constraints in create table:**

- **not null**
- **primary key** (*column_name*$_1$,…, *column_name*$_n$)
- **foreign key** (*column_name*$_1$) **references** *table_name*(*column_name*$_n$)

- **primary key** declaration on an attribute automatically ensures **not null**

# Create table Example

```
Create table instructor
(
        ID          char(5),
        name        varchar(20) not null,
        dept_name   varchar(20),
        salary      numeric(8,2),
        primary key (ID),
        foreign key(dept_name) references department (dept_name)
);
```

Attributes of primary key and foreign key must be defined first in the attribute list.

For instance, in the above example we first need to define the attributes ***ID char (5), dept_name varchar(20)*** first before we define PK and FK.

# Updates to table structure

- **Drop Table**
  - **drop table** *r* *//this will delete the table and its data*

- **Alter**
  - **alter table** *r* **add** *A D* *//this command to add a new attribute to the table*
    - where *A* is the name of the attribute
    - *r is the relation name*
    - and *D* is the domain of *A (data type)*
  - **alter table** *r* **drop** *A* *//this command to remove an existing attribute*
    - where *A* is the name of an attribute
    - *r is the relation name*

# Drop table Ex.

To drop a table:

**Drop table** *instructor*;

To drop a table and its constraints: (when the table has a relationship with other tables via foreign key constraints)

**Drop table** *department* **cascade constraints;**

Try dropping table department without **cascade constraints**!

# Alter table syntax

To add a column

```
Alter table table_name add column_name datatype;
```

To drop a column:

```
Alter table table_name drop column column_name;
```

To modify a column:

```
Alter table table_name modify column_name datatype;
```

# SQL Summary (DDL)

**Create table** *table_name* **(**

*column_name1* *datatype(size)***,**

*column_name2* *datatype(size)***,**

*column_name3* *datatype(size)***,**

*……*

***primary key*** *(column_name$_1$,…, column_name$_n$),*

***foreign key*** *(column_name$_1$)* ***references*** *anotherTable_name(column_name$_n$)*

**);**

**Drop table** *table_name***;**

**Drop table** *table_name* **cascade constraints;**

**Alter table** *table_name* **add** *column_name* *datatype***;**

**Alter table** *table_name* **drop column** *column_name***;**

**Alter table** *table_name* **modify** *column_name* *datatype***;**