



# Database System Lab (CSE 3103)

Session 02

Nazmus Sakib, Assistant Professor, Dept. of CSE, AUST

# Best Practice

- Table: Use upper case and singular form in table name. Not plural.
- Column: Use the pascal notation. E.g. “StudentId”
- Primary Key:
  - If the table name is “Course”, name the primary key column “CourseId”
  - Always use integer and identity(1,1) for primary key. Use Unique constraint for other columns that needs to be unique
- Specify required columns NOT NULL – which columns that need to have data or not.

# INSERT data into Database

- The INSERT INTO statement is used to insert a new row in a table.
- Please remember Values are in **single inverted commas**'.
- The first form doesn't specify the column names where the data will be inserted, only their values:

```
INSERT INTO table_name  
VALUES (value1, value2, value3,...)
```

# INSERT data into Database

- The second form specifies both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2,  
column3,...)  
VALUES (value1, value2, value3,...)
```

## INSERT data into PERSON table

<b>Id</b>	<b>Name</b>	<b>Address</b>	<b>Phone</b>
<b>1</b>	Hansen	Dhaka	016118899
<b>2</b>	Svendson	Khulna	018234568
<b>3</b>	Pettersen	Sylhet	017234569
<b>4</b>	Nilsen	Dhaka	01919876
<b>5</b>	Tjessem	Rangpur	01552765

# SQL Constraints

- PRIMARY KEY – A Unique value that must contain in a table. Each table should have a primary key.
- NOT NULL – the attribute which may contain value or not.
- UNIQUE – identify each record in the database uniquely.
- FOREIGN KEY – It is the primary key of another table , use as attribute to a table from another table.
- CHECK – check the value with the range or number
- DEFAULT – set like a fixed data
- IDENTITY or AUTO INCREMENT – automatic every time recorded

# CREATE TABLE Statement

```
CREATE TABLE CUSTOMER
(
  CustomerId int IDENTITY(1,1) PRIMARY KEY,
  LastName varchar(50) NOT NULL,
  FirstName varchar(50) NOT NULL,
  AreaCode int NULL,
  Address varchar(200) NULL,
  Birthday date NOT NULL,
  Salary decimal(7,2) NULL,
  Phone varchar(11) NOT NULL,
)
```



# INSERT data into Customer table

CustomerId	LastName	FirstName	AreaCode	Address	Birthday	Salary	Phone
1	Hansen	Ola	1203	Dhaka	2000-12-18	1020.54	016118899
2	Svendson	Tove	1212	Khulna	1997-08-13		018234568
3	Pettersen	Kari		Sylhet	1992-08-28	1234.56	017234569
4	Nilsen	Tove	1243	Dhaka	1995-05-23		01919876
5	Tjessem	Jakob		Rangpur	2010-11-12	43215.22	01552765

# FOREIGN KEY Statement

```
CREATE TABLE ORDER
```

```
(  
  OrderId int IDENTITY (100, 1) PRIMARY KEY,  
  CustomerId int NOT NULL FOREIGN KEY REFERENCES CUSTOMER (CustomerId),  
  OrderDate date NULL,  
  OrderAmount money NOT NULL,  
)
```

# SQL Date Data Types

- **DATE** - format YYYY-MM-DD
- **DATETIME** - format: YYYY-MM-DD HH:MM:SS
- **SMALLDATETIME** - format: YYYY-MM-DD HH:MM:SS
- **TIMESTAMP** - format: a unique number

## INSERT data into **Order** table

OrderId	CustomerID	OrderDate	OrderAmount
100	1	2018-12-18	102.45
101	2	2018-08-13	5039
102	1	2018-08-28	1234.90
103	3	2018-05-23	
104	6	2018-11-12	4315.22

# SELECT Statement

- The SELECT statement is used to select data from a database.

```
SELECT column_name(s)  
FROM table_name
```

Or

```
SELECT * FROM table_name
```

# SELECT Statement

- select the content of the columns named "LastName" and "FirstName" from the table named Customer.
- select all the content of the table named Customer.

# SELECT DISTINCT Statement

- In a table, some of the columns may contain duplicate values. This is not a problem, however, sometimes you will want to list only the different (distinct) values in a table.
- The **DISTINCT** keyword can be used to return only distinct (different) values.

# SELECT DISTINCT Statement

- Syntax

```
SELECT DISTINCT column_name(s)  
FROM table_name
```

- Now we want to select only the distinct values from the column named "OrderDate" from the table named Order