MySQL Integrity Constraints

(LAB Lecture)

Department of Information Technology

Database Systems II (IT216)

Spring 2025-2026

Week 3 – February 17, 2026

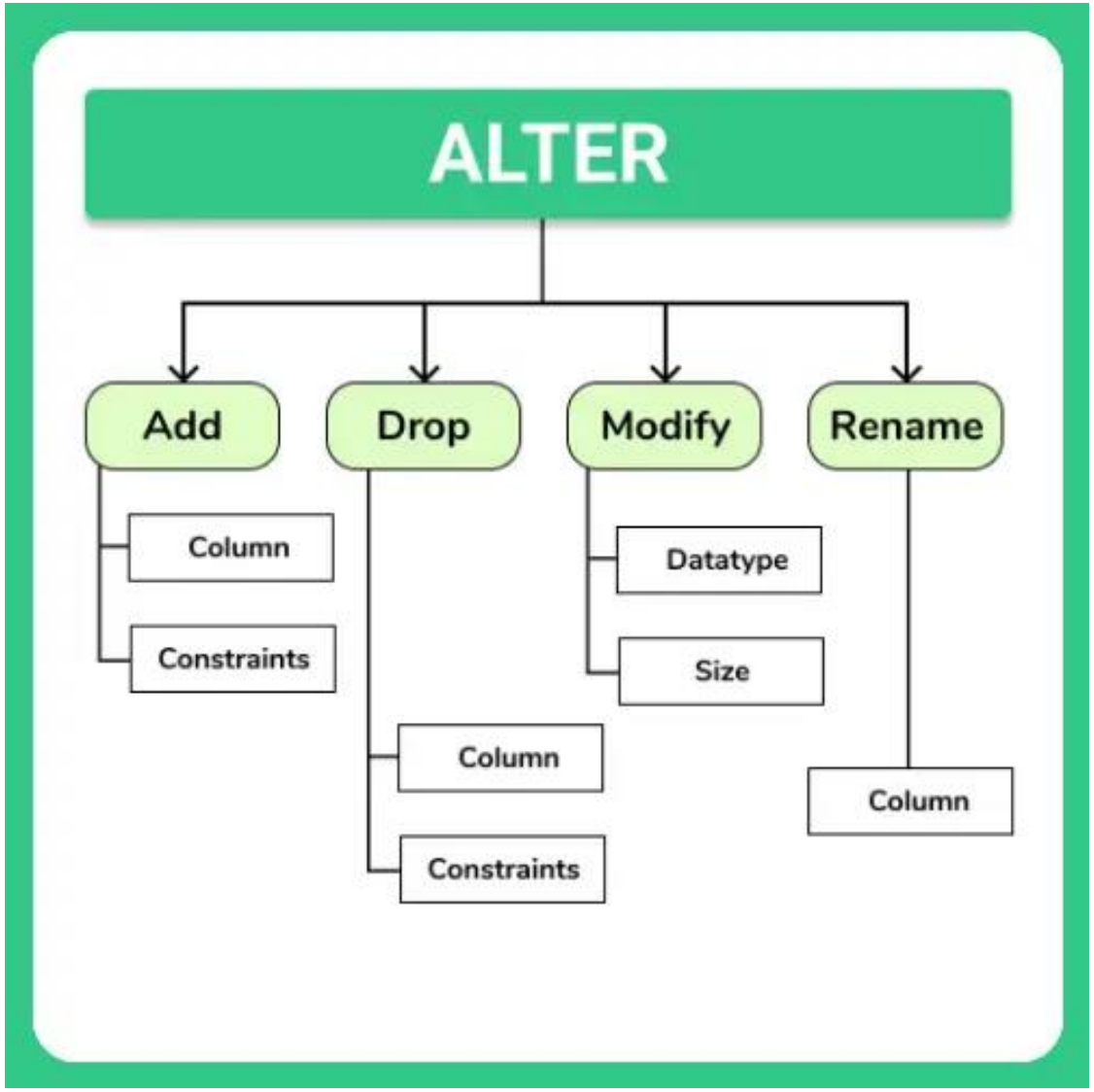
Lecturer: Soma Soleimanzadeh

Contents

- MySQL Integrity Constraints
- Alter Table
    - Add a Constraint in CREATE TABLE
    - Add a Constraint in ALTER TABLE

ALTER TABLE for Adding/Dropping Constraints

- **Integrity Constraints** are **rules** to ensure that the data in the database is accurate, consistent and reliable.
- Two ways to specify constraints:
  - while creating a table in **CREATE TABLE**
  - after table creation in **ALTER TABLE**



Most Common MySQL Constraints

- **NOT NULL**
- **AUTO_INCREMENT**
- **UNIQUE**
- **CHECK**
- **DEFAULT**
- **PRIMARY KEY**
- **FOREIGN KEY**

Let's Try Specifying Some Constraints

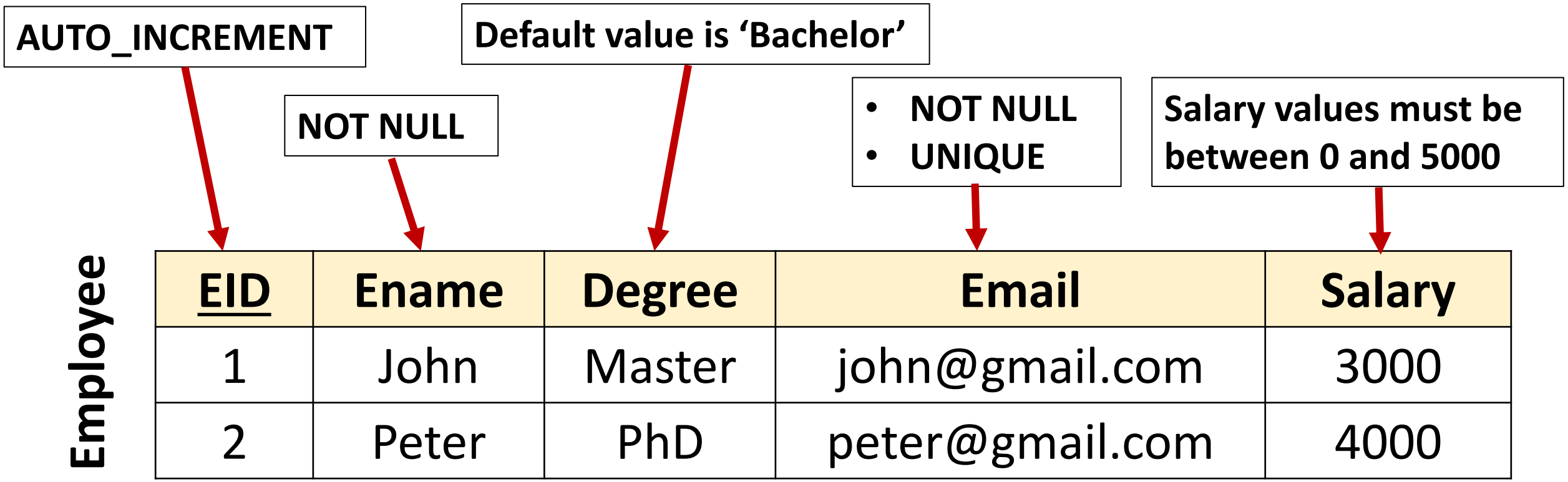
- In the next slides, we will follow to scenarios to practice specifying integrity constraints in tables:

  - **Scenario 1 –** Creating a table and specifying constraints in **CREATE TABLE** Statement.

  - **Scenario 2 –** Creating a table without any constraints except for primary key, then adding constraints in **ALTER TABLE** Statement.

Scenario 1 – Adding Constraints in CREATE TABLE

Create the following table, considering all mentioned constraints in **CREATE TABLE** Command.



Scenario 1 – CREATE TABLE

```
create table employee
(EID int auto_increment,
EName varchar(50) NOT NULL,
Degree enum('Bachelor','Master','PhD') default 'Bachelor',
Email varchar(60) UNIQUE NOT NULL,
Salary int check(Salary>=0 and Salary<=5000),
primary key(EID));
```

Scenario 1 – Inserting Some Data into Table

General Syntax of Inserting Data into a Table:

INSERT INTO table_name(column_names) **VALUES** (values_for_columns)

EID	Ename	Degree	Email	Salary

Insert into employee(Ename, Degree, Email, Salary)
Values (*Null*, 'Master', 'karez@gmail.com', 3000);

Error! → Because **NOT NULL** constraint of **Ename** is violated.

Scenario 1 – Inserting Some Data into Table

EID	Ename	Degree	Email	Salary
1	Ahmed	**Bachelor**	ahmed@gmail.com	4000

Insert into employee(Ename, Email, Salary)
Values ('Ahmed', 'ahmed@gmail.com', 4000);

The **Degree** value is not specified. Because the **Degree** column has **Default** constraint, it sets to the default value (**Bachelor**).

Scenario 1 – Inserting Some Data into Table

EID	Ename	Degree	Email	Salary
1	Ahmed	Bachelor	ahmed@gmail.com	4000

Insert into employee(Ename, Degree, Email, Salary)
Values ('Sara', 'PhD', 'ahmed@gmail.com', 3500);

Error! → Because **UNIQUE** constraint of the **Degree** column is violated that says repeated values are not allowed in **Email**.

Insert into employee(Ename, Degree, Email, Salary)
Values ('Didar', 'Bachelor', 'didar@gmail.com', 7000);

Error! → Because **CHECK** constraint of Salary column is violated that says Salary must be between 0 and 5000.

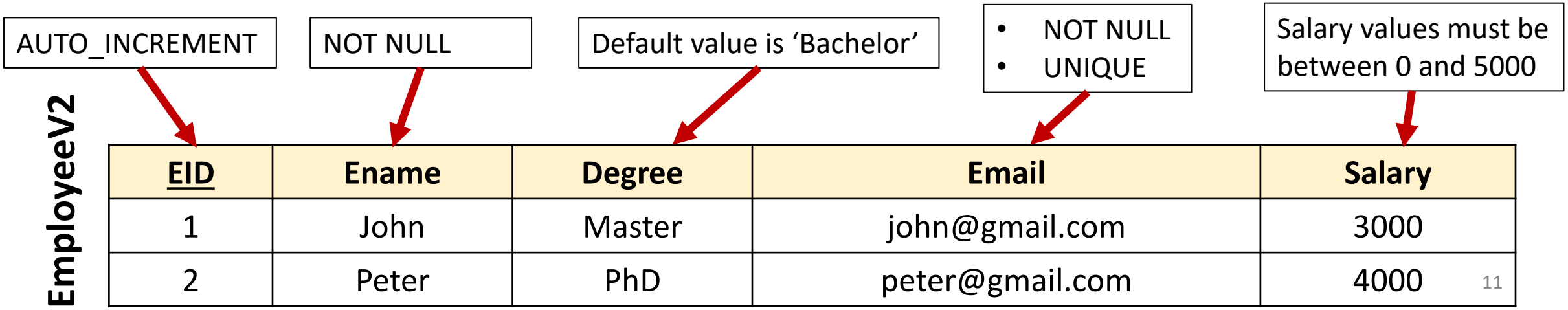
Scenario 2 – Adding Constraints in ALTER TABLE

1. Create the following table, without specifying any constraints, except for primary key.

EmployeeV2

EID	**Ename**	**Degree**	**Email**	**Salary**
1	John	Master	john@gmail.com	3000
2	Peter	PhD	peter@gmail.com	4000

2. After the table was created, use **ALTER TABLE** to add the following constraints:



Scenario 2 – CREATE TABLE without Constraints

```
create table employeeV2
(EID int,
EName varchar(50),
Degree enum('Bachelor','Master','PhD'),
Email varchar(60),
Salary int,
primary key(EID));
```

Scenario 2 – Adding Constraints in ALTER TABLE

```sql
alter table employeeV2
modify EID int auto_increment;

alter table employeeV2
modify EName varchar(50) NOT NULL;

alter table employeeV2
alter Degree set default 'Bachelor';
```

```sql
alter table employeeV2
modify Email varchar(60) NOT NULL;

alter table employeeV2
add unique(Email);

alter table employeeV2
add check(Salary>=0 and Salary<=5000);
```

Scenario 2 – Inserting Some Data into Table

- You can test all constraints by entering data into the **employeeV2** table, similar to entering data into the **employee** table in scenario 1.
- The results you get are the same for both scenarios. It means that even if we forget to add a constraint to a column in the CREATE TABLE, we can add it later in the ALTER TABLE command.