Week 4 Tutorial Notes

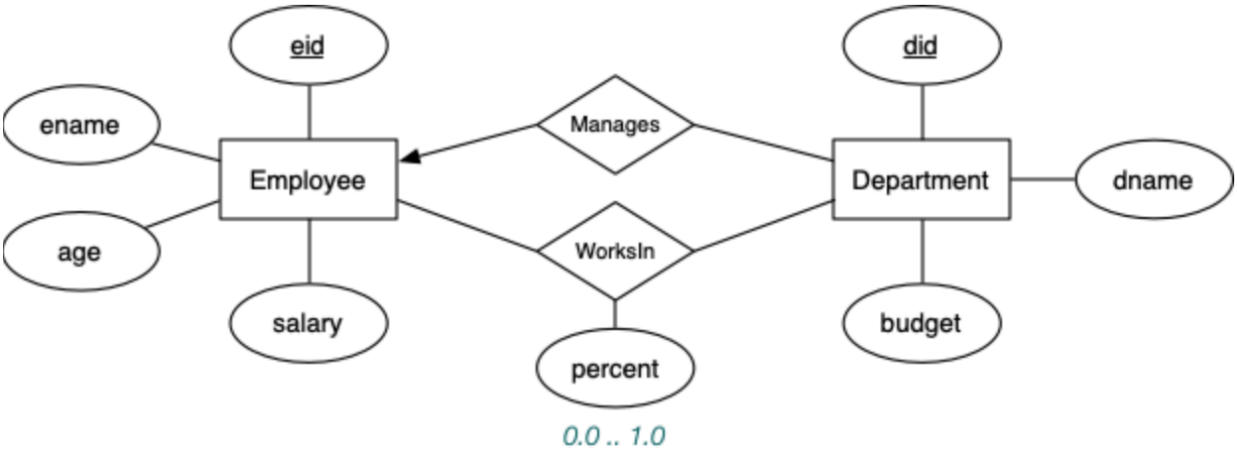
▼ Agenda

- Q1-10 → Constraints on create tables
- Q11 → SQL DELETE
- Q12-23 → SQL SELECT

Pre-Tute

- Quiz 3 due on Friday
- Assignment 1 due 9pm next Friday (Week 5)
- Help session... TBD

SQL Data Definition Language - a recap



▼ SQL CREATE TABLE

```
create table Employees (
      eid     integer primary key,
      ename   text,
      age     integer,
      salary  real,
      primary key (eid)
);
create table Departments (
      did     integer primary key,
      dname   text,
      budget  real,
      manager integer references Employees(eid)
);
create table WorksIn (
      eid     integer references Employees(eid),
      did     integer references Departments(did),
      percent real,
      primary key (eid,did)
);
```

SQL ALTER TABLE

▼ Changing column names

```
ALTER TABLE Departments
ALTER COLUMN dname department_name;
```

▼ Dropping columns

```
ALTER TABLE Departments
DROP COLUMN budget;
```

▼ Adding columns

```
ALTER TABLE Employees
ADD Email VARCHAR(255);
```

SQL Constraints

Describes the nature of an attribute/column's data:

▼ Data types

```
NUMERIC
INTEGER / INT
BOOLEAN

TEXT
CHAR(size)
VARCHAR(size)
ENUM(val1, val2, val3, ...)

DATE
TIME
```

▼ CHECK

```
CREATE TABLE Employees (
    age INTEGER CHECK (age >= 18 AND age <= 65),
    salary NUMERIC CHECK (salary > 0)
);
```

▼ NOT NULL

```
CREATE TABLE Students (
    student_id INTEGER NOT NULL,
```

```
    name VARCHAR(100) NOT NULL
);
```

▼ DEFAULT

```
CREATE TABLE Products (
    stock INT DEFAULT 0
);
```

▼ UNIQUE

```
CREATE TABLE Users (
    email VARCHAR(255) UNIQUE
);
```

▼ PRIMARY KEY

```
CREATE TABLE Books (
    bId CHAR(13) PRIMARY KEY
);
```

▼ FOREIGN KEY

```
CREATE TABLE Orders (
    customer INT REFERENCES Customers(customer_id)
);
```

❓ Tutorial Q1-7

SQL DELETE

In a foreign key constraint, what happens to the child table when a referenced parent row is deleted?

▼ ON DELETE RESTRICT (or ON DELETE): Disallow the delete

```
CREATE TABLE Orders (
    customer_id INT,
    FOREIGN KEY (customer_id) REFERENCES Customers(id)
        ON DELETE RESTRICT
);
```

▼ ON DELETE CASCADE: Delete all child rows that refer to it

```
CREATE TABLE Orders (
    customer_id INT,
    FOREIGN KEY (customer_id) REFERENCES Customers(id)
        ON DELETE CASCADE
);
```

▼ ON DELETE SET DEFAULT: The foreign key in child rows are set to the default value

```
CREATE TABLE Orders (
    customer_id INT DEFAULT 0,
    FOREIGN KEY (customer_id) REFERENCES Customers(id)
        ON DELETE SET DEFAULT
);
```

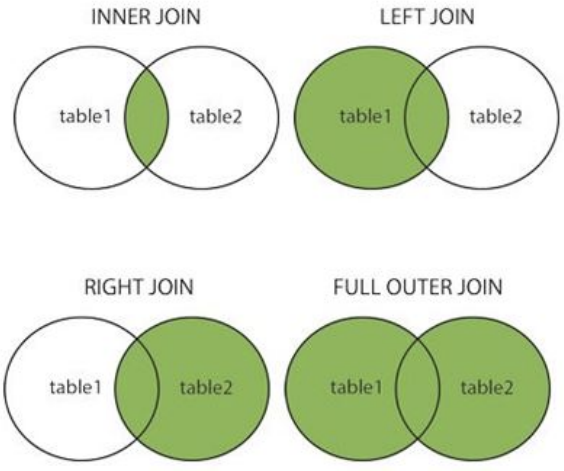
❓ Tutorial Q8-11

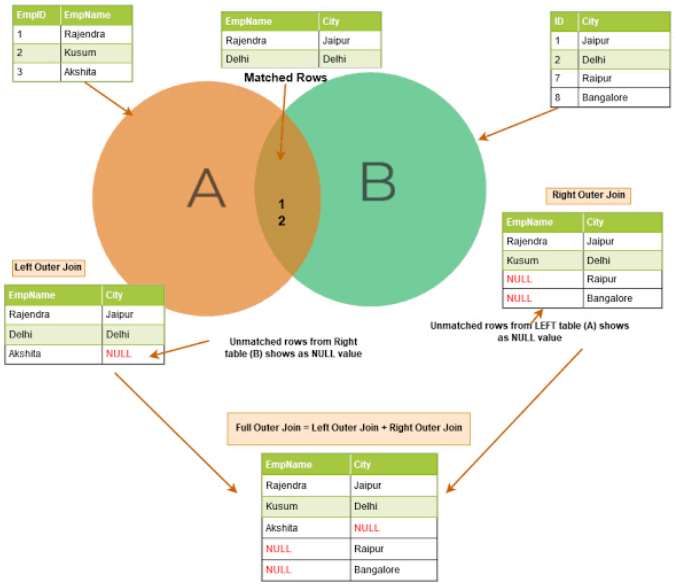
SQL SELECT

```
SELECT
   att1, att2, ...
FROM
   table_name
WHERE
   condition
<ORDER BY, GROUP BY>
```

SQL JOIN

```
SELECT
   att1, att2, ...
FROM
   tableA
JOIN
   tableB ON (tableA.id = tableB.Aid)
```





SQL Aggregates

Generally a select statement returns a list of tuples. We may want to aggregate them:

- count
- sum
- min
- max
- avg
- string_agg

▼ Example: Count of all employees with salary > 1000

```
SELECT
    count(eid)
FROM
    Employees
```

```
WHERE
   salary > 1000
```

▼ Example: Average salary

```
SELECT
   avg(salary)
FROM
   Employees
```

SQL GROUP BY

Group tuples with distinct attributes. Any attributes that are in the SELECT but **not** in GROUP BY **must be aggregated**

▼ Example: For each age, shows the number of employees, and the average salary

```
SELECT
   age,
   count(eid),
   avg(salary)
FROM
   Employees
GROUP BY
age
```

SQL CREATE VIEW

Stores a ‘snapshot’ of an instance of a table

```
CREATE VIEW view_name AS
SELECT
   *
FROM
   table_name
WHERE
   (conditions)
;
```