

# Queries in SQL – Product and Join

Hui Chen <sup>a</sup>

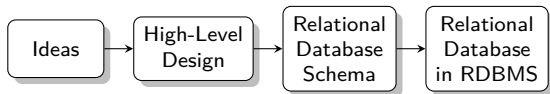
<sup>a</sup>CUNY Brooklyn College, Brooklyn, NY, USA

March 3, 2022

# Outline

- 1 Introduction to SQL
- 2 Queries in SQL
  - Products
  - Joins
- 3 Projection and Join
  - Bag Union, Intersection, and Difference
- 4 Assignment

# Overview



# Outline

- 1 Introduction to SQL
- 2 Queries in SQL
  - Products
  - Joins
- 3 Projection and Join
  - Bag Union, Intersection, and Difference
- 4 Assignment

# Introduction to SQL

SQL (pronounced as “sequel”) is the principal language used to describe and manipulate relational database, and has several aspects:

- ▶ Data definition language (DDL).
  - ▶ SQL includes commands to create database objects such as tables, indexes, and views, as well as commands to define access rights to those database objects.
  - ▶ Topics of this lecture: SQL commands to create database tables (relations)
- ▶ Data manipulation language (DML).
  - ▶ SQL includes commands to insert, update, delete, and retrieve data within the database tables.
- ▶ Transaction control language (TCL).
  - ▶ The DML commands in SQL are executed within the context of a transaction.
- ▶ Data control language (DCL).
  - ▶ Data control commands are used to control access to data objects.

# Outline

- 1 Introduction to SQL
- 2 Queries in SQL
  - Products
  - Joins
- 3 Projection and Join
  - Bag Union, Intersection, and Difference
- 4 Assignment

# Queries to SQL

A SQL can be understood as a relational algebra query. We discussed

- ▶ Selection
- ▶ Projection

These queries involve only a single relation. How about the queries involving more than one relation?

- ▶ Products
- ▶ Joins

# Products

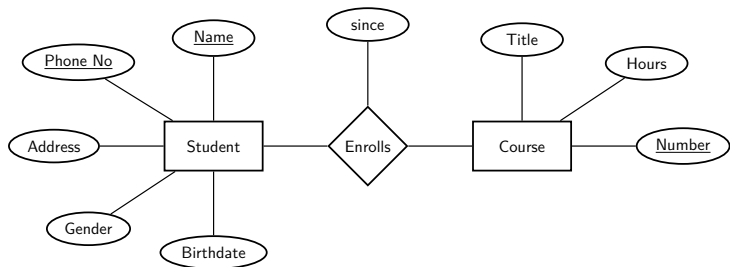
$$R3 := R1 \times R2 \quad (1)$$

```
SELECT *  
FROM R1, R2
```



# Let's Consider Our Example on Students and Courses ...

Consider the following database model



whose relational database schemas are,

```
Students(name:string, phone:string, address:string,
gender:string, birthdate:date)
```

```
Courses(name:string, title:string, number, hours:integer)
```

```
Enrollment(sname:string, sphone:string, cnumber:string,
since:datetime)
```

## Products in SQL: Example

*Results := Students  $\times$  Enrollment* (2)

```
SELECT *  
FROM Students, Enrollment;
```

## $\theta$ -Join

$$R3 := R1 \bowtie_C R2 \quad (3)$$

```
SELECT *  
FROM R1, R2  
WHERE C
```

or more explicitly

```
SELECT *  
FROM R1 INNER JOIN R2  
ON C
```

```
SELECT *  
FROM R1 INNER JOIN R2  
WHERE C
```

“INNER JOIN”? Is there an “OUTER JOIN”?

# Natural Join

$$R3 := R1 \bowtie R2 \quad (4)$$

```
SELECT *  
FROM R1 NATURAL JOIN R2
```

## $\theta$ -Join: Example

*Results := Students*

$\bowtie$  *Students.name=Enrollments.sname AND Students.phone=Enrollment.sphone*  
*Enrollments* (5)

```
SELECT *
FROM Students, Enrollment
WHERE Students.phone = Enrollment.sphone
       AND Students.name = Enrollment.sname;
```

or more explicitly

```
SELECT *
FROM Students INNER JOIN Enrollment
WHERE Students.phone = Enrollment.sphone
       AND Students.name = Enrollment.sname;
```

What question does this query answer?

## Natural-Join: Example

*Results* := *Students* ⋈ *Enrollments* (6)

```
SELECT *  
FROM Students NATURAL JOIN Enrollment
```

For the schemas we have, what does it really do?

# Outline

- 1 Introduction to SQL
- 2 Queries in SQL
  - Products
  - Joins
- 3 Projection and Join
  - Bag Union, Intersection, and Difference
- 4 Assignment

# Combining Project with Join

$$R3 := \pi_L(R1 \bowtie_C R2) \quad (7)$$

```
SELECT L
FROM R1, R2
WHERE C
```

or more explicitly

```
SELECT L
FROM R1 INNER JOIN R2
ON C
```

```
SELECT L
FROM R1 INNER JOIN R2
WHERE C
```



## Projection and $\theta$ -Join: Example

$Results := \pi_{Enrollment.cidnum}(Students$

$\bowtie_{Students.name=Enrollments.sname \wedge Students.phone=Enrollment.sphone}$   
 $Enrollments) \quad (8)$

```
SELECT Enrollment.cidnum
FROM Students, Enrollment
WHERE Students.phone = Enrollment.sphone
       AND Students.name = Enrollment.sname;
```

or more explicitly

```
SELECT Enrollment.cidnum
FROM Students INNER JOIN Enrollment
WHERE Students.phone = Enrollment.sphone
       AND Students.name = Enrollment.sname;
```

What question does this query answer?

## Union, Intersection, and Difference

- ▶ (Union) For  $\cup$ , use UNION
- ▶ (Intersection) For  $\cap$ , use INTERSECT
- ▶ (Difference) for  $-$ , use EXCEPT

Note: MariaDB supports INTERSECT and EXCEPT from version 10.3.0

# Outline

- 1 Introduction to SQL
- 2 Queries in SQL
  - Products
  - Joins
- 3 Projection and Join
  - Bag Union, Intersection, and Difference
- 4 Assignment

# Assignment

Let's work on an assignment using paper and pencil/pen ...