SQL: Intermediate Queries

Database Systems

Department of Computer Science, CBNU
Prof. Nasridinov Aziz (aziz@chungbuk.ac.kr)

In the Last Lecture

- Data Definition Language (DDL)
  - Notation for defining the database schema
    - Used to create and modify the structure of your relations and database objects
  - Statements
    - CREATE
    - DROP
    - ALTER
    - TRUNCATE

In the Last Lecture

- MySQL constraints
  - SQL constraints are used to specify rules for the data in a relation
  - MySQL constraints
    - **NOT NULL**
    - **PRIMARY KEY**
    - **FOREIGN KEY**
    - **UNIQUE**
    - **CHECK**
    - **DEFAULT**

In the Last Lecture

- Data Manipulation Language (DML)
  - A language for searching, modifying, inserting and deleting the desired data in the database
    - DML also known as query language
  - Statements
    - INSERT
    - UPDATE
    - DELETE
    - SELECT

In the Last Lecture

- The SELECT statement
  - Used to select/retrieve tuples from a database

  - Syntax

SELECT	[**DISTINCT**] attribute(s)	(1)	Compulsory
FROM	relation(s)	(2)	
[**WHERE**	condition]	(3)	Optional
[**GROUP BY**	attribute(s)]	(4)	
[**HAVING**	condition]	(5)	
[**ORDER BY**	attribute(s)]	(6)	
[**LIMIT**	number];	(7)	

In the Last Lecture

- The GROUP BY statement
  - Query: Group the employees by department and calculate avg and max salaries

```
SELECT    dno, AVG(salary) as avgsal, MAX(salary) as maxsal
FROM      employee
GROUP BY  dno;
```

EMPLOYEE

EMPNO	EMPNAME	TITLE	MANAGER	SALARY	DNO
3426	박영권	과장	4377	3000000	1
1365	김상원	사원	3426	1500000	1
2106	김창섭	대리	1003	2500000	2
1003	조민희	과장	4377	3000000	2
4377	이성래	사장	∧	5000000	2
3011	이수민	부장	4377	4000000	3
3427	최종철	사원	3011	1500000	3

Grouped

DNO	AVGSAL	MAXSAL
1	2250000	3000000
2	3500000	5000000
3	2750000	4000000

Table of Contents

Part 1

INTERMEDIATE DDL

1. Intermediate DDL

- MySQL constraints
  - SQL constraints are used to specify rules for the data in a relation
  - MySQL constraints
    - **NOT NULL**
    - **PRIMARY KEY**
    - **FOREIGN KEY**
    - **UNIQUE**
    - **CHECK**
    - **DEFAULT**

→ rules → constraints

1. Intermediate DDL

재입력 불가능.

- UNIQUE constraint
  - Ensures that all values in an attribute are different

UNIQUE

1+ 1

  - UNIQUE constraint vs. PRIMARY KEY constraint
    - Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for an attribute or set of attributes

같은 행에 같은 값 여러개 조회 불가능.

    - A PRIMARY KEY constraint automatically has a UNIQUE constraint

    - However, you can have many UNIQUE constraints per relation, but only one PRIMARY KEY constraint per relation

유니크 = 여러개 가능.
프라이머리 = 오직 1개.

1. Intermediate DDL

- DEFAULT constraint
  - Used to set a default value for an attribute
    - The default value will be added to all new tuples, if no other value is specified

- CHECK constraint
  - The CHECK constraint is used to limit the value range that can be placed in a column
    - It will allow only certain values for this column

1. Intermediate DDL

- Example for MySQL constraints

default constraint

```
CREATE TABLE employee(
        empno           int             NOT NULL,
        empname         varchar(45),
        title           varchar(45)                 DEFAULT '사원',
        manager         int,
        salary          int,
        dno             int,
        CONSTRAINT PK_Employee PRIMARY KEY (empno),
        CONSTRAINT FK_Employee_Manager
        FOREIGN KEY (manager) REFERENCES employee(empno),
        CONSTRAINT FK_Department_Employee
        FOREIGN KEY (dno) REFERENCES department(deptno),
        CONSTRAINT UQ_empname UNIQUE (empname),
        CONSTRAINT CH_salary CHECK (salary < 6000000),
        CONSTRAINT CH_dno CHECK (dno IN (1, 2, 3, 4, 5, 6))
);
```

Part 2

INTERMEDIATE DML

2. Intermediate DML

❖ Join Queries

- Using the EMPLOYEE table, find the department in which the employee named '김창섭' works

EMPLOYEE

EMPNO	EMPNAME	TITLE	MANAGER	SALARY	DNO
2106	김창섭	대리	1003	2500000	2
3426	박영권	과장	4377	3000000	1
3011	이수민	부장	4377	4000000	3
1003	조민희	과장	4377	3000000	2
3427	최종철	사원	3011	1500000	3
1365	김상원	사원	3426	1500000	1
4377	이성래	사장	∧	5000000	2

- Query

```
SELECT      *
FROM        employee
WHERE       empname = '김창섭';
```

2. Intermediate DML

❖ Join Query

- Combines tuples from two or more relations based on a common attribute related column → Foreign key

EMPLOYEE

EMPNO	EMPNAME	TITLE	MANAGER	SALARY	DNO
2106	김창섭	대리	1003	2500000	2
3426	박영권	과장	4377	3000000	1
3011	이수민	부장	4377	4000000	3
1003	조민희	과장	4377	3000000	2
3427	최종철	사원	3011	1500000	3
1365	김상원	사원	3426	1500000	1
4377	이성래	사장	∧	5000000	2

DEPARTMENT

DEPTNO	DEPTNAME	FLOOR
1	영업	8
2	기획	10
3	개발	9
4	총무	7

2. Intermediate DML

- Join Query
  - Syntax

```
SELECT table1.column1, table2.column2
FROM table1
JOIN table2 ON
        table1.common_column = table2.common_column;
```

skip X.

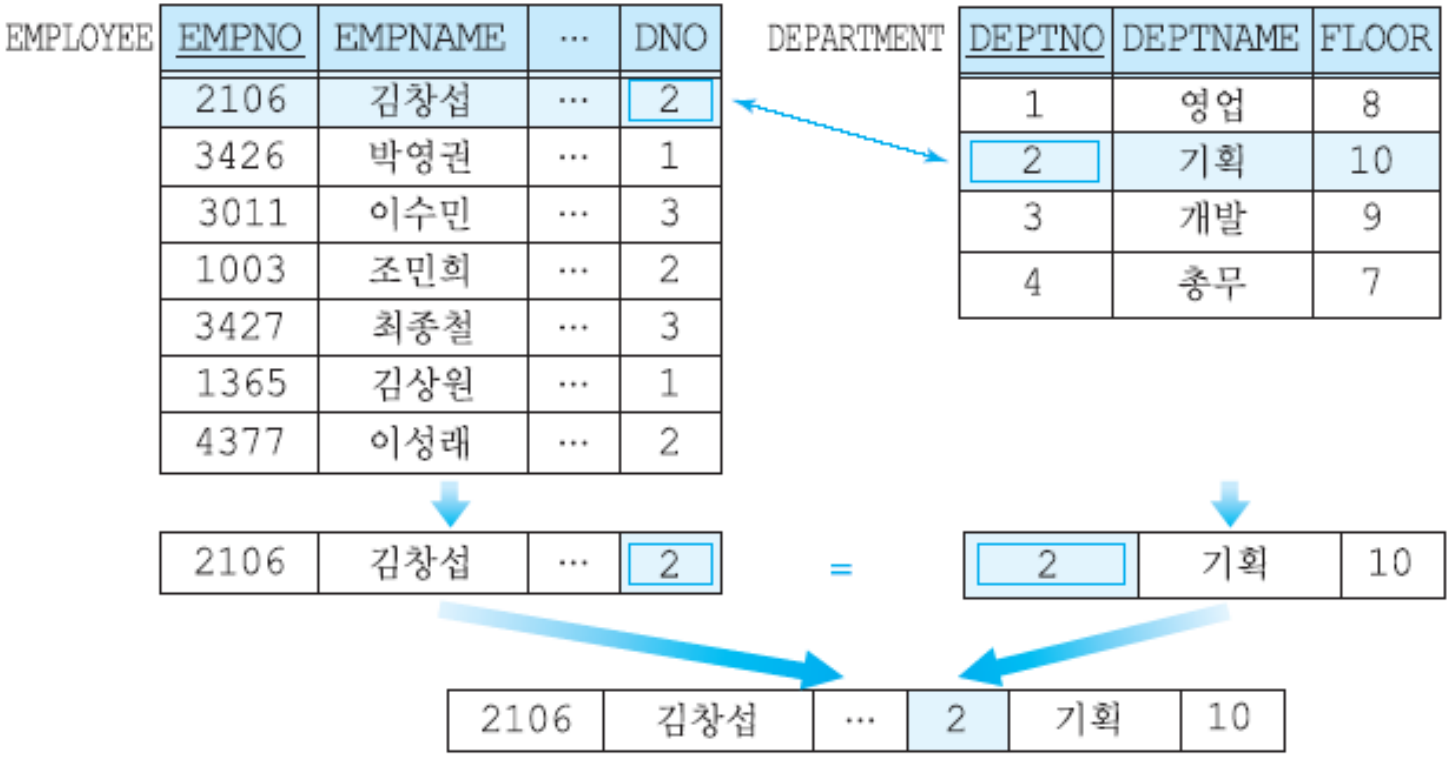
Join Condition

  - Example Query

```
SELECT *
FROM employee
JOIN department ON employee.dno = department.deptno
WHERE empname = '김창섭';
```

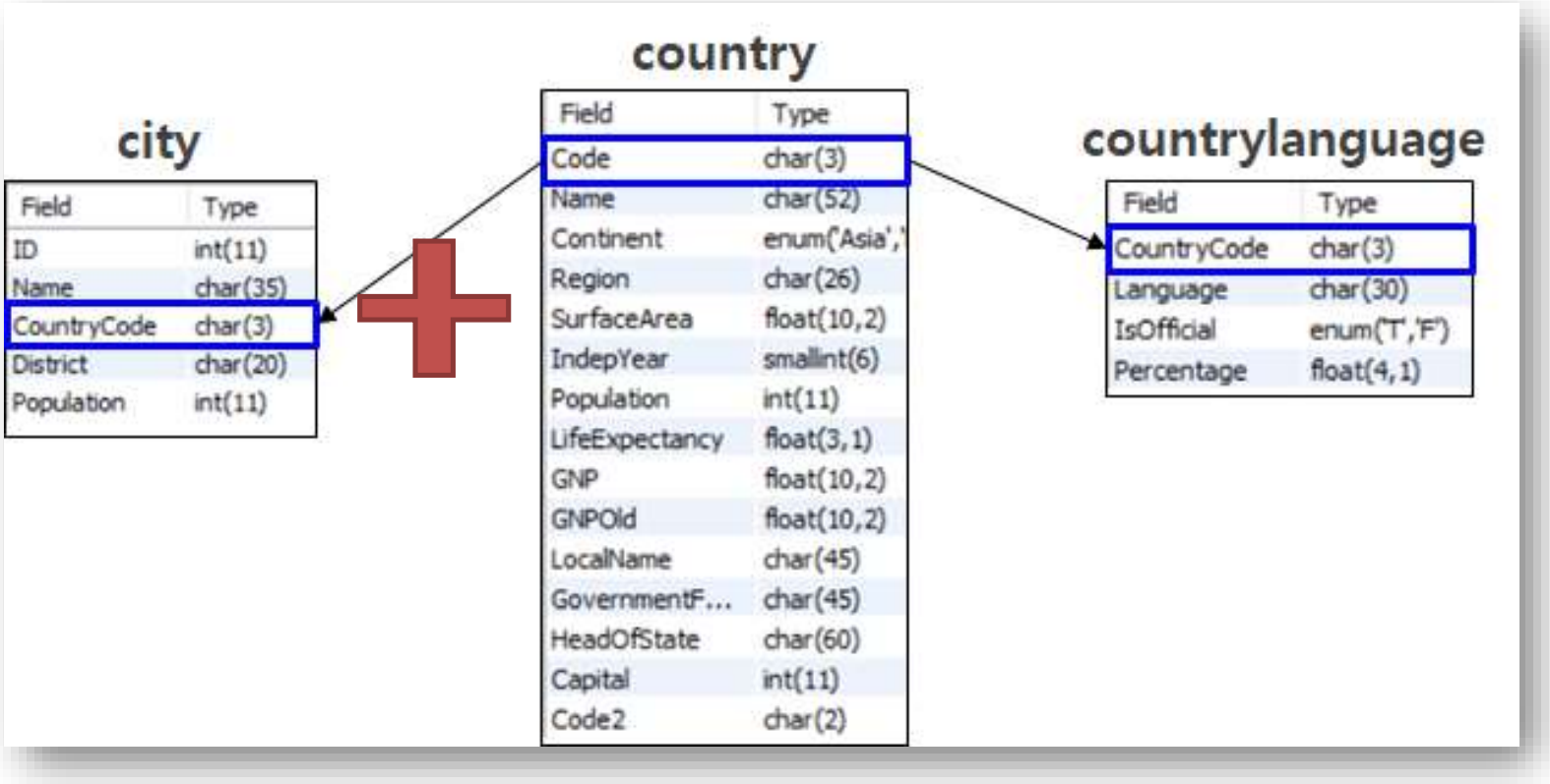
2. Intermediate DML

- Join Query
  - How it works



2. Intermediate DML

- Join Query (Example)
  - Q1: Show the countries and their cities



2. Intermediate DML

- Join Query (Example)
  - Q1: Show the countries and their cities

    SELECT CountryCode, Name

    FROM city

    ORDER BY CountryCode;

  - Explanation
    - You can find the countries and their cities from CITY table by simply looking through CountryCode

2. Intermediate DML

- Join Query (Example)
  - Q1: Show the countries and their cities
  - Result

CountryCode	Name
ABW	Oranjestad
AFG	Kabul
AFG	Qandahar
AFG	Herat
AFG	Mazar-e-Sharif
AGO	Luanda
AGO	Huambo
AGO	Lobito
AGO	Benguela
AGO	Namibe
AIA	South Hill

2. Intermediate DML

❖ Join Query (Example)

- Q1: Show the countries and their cities (**Better Query!**)

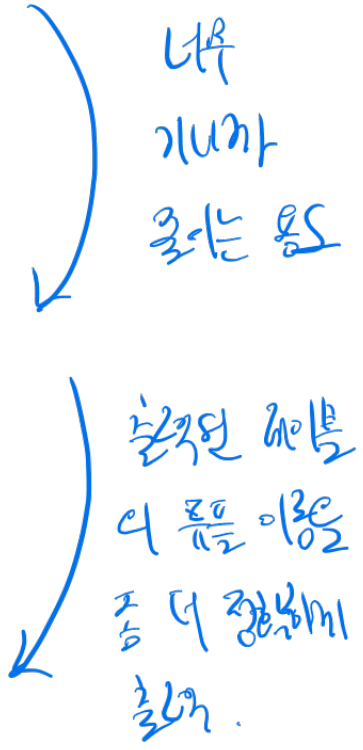
```
SELECT country.Name, city.Name
FROM city
JOIN country ON city.CountryCode = country.Code;
```

- Q1: Show the countries and their cities (**EVEN Better Query!**)

```
SELECT C.Name, T.Name
FROM city T
JOIN country C ON T.CountryCode = C.Code;
```

- Q1: Show the countries and their cities (**THE BEST!**)

```
SELECT C.Name AS Country_Name, T.Name AS City_Name
FROM city T
JOIN country C ON T.CountryCode = C.Code;
```



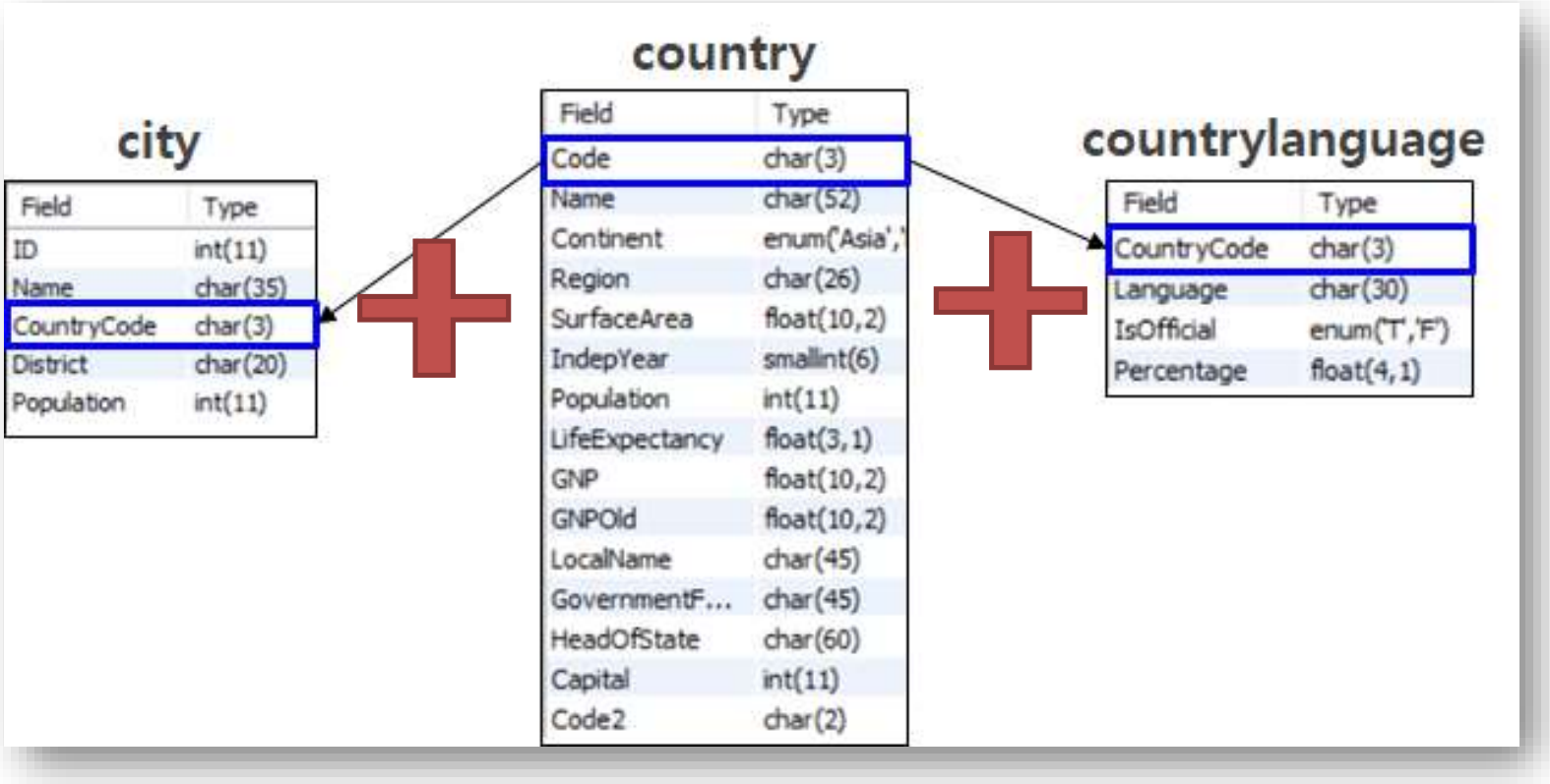
2. Intermediate DML

- Join Query (Example)
  - Q1: Show the countries and their cities
  - Result

Country_Name	City_Name
Aruba	Oranjestad
Afghanistan	Kabul
Afghanistan	Qandahar
Afghanistan	Herat
Afghanistan	Mazar-e-Sharif
Angola	Luanda
Angola	Huambo
Angola	Lobito
Angola	Benguela
Angola	Namibe
Anguilla	South Hill

2. Intermediate DML

- Join Query (Example)
  - Q2: Language used in Berlin and where it is located?



2. Intermediate DML

- Join Query (Example)
  - Q2: Language used in Berlin and where it is located?

  - Query

```
SELECT T.Name AS City,
       C.Name AS Country,
       L.Language,
       L.Percentage
FROM city T
JOIN country C ON T.CountryCode = C.Code
JOIN countrylanguage L ON C.Code = L.CountryCode
WHERE T.Name = 'Berlin' AND L.IsOfficial = 'T';
```

2. Intermediate DML

- Join Query (Example)
  - Q2: Language used in Berlin and where it is located?
  - Result

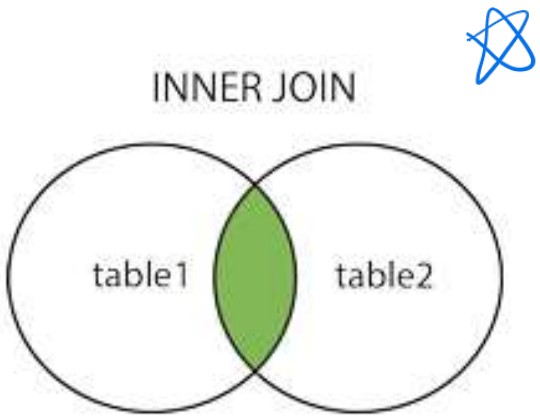
City	Country	Language	Percentage
Berlin	Germany	German	91.3

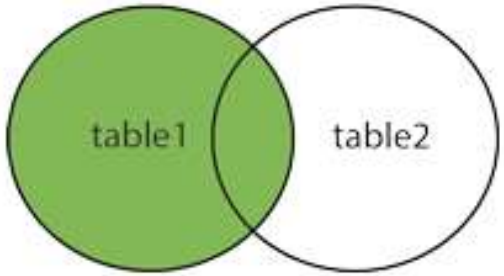
2. Intermediate DML

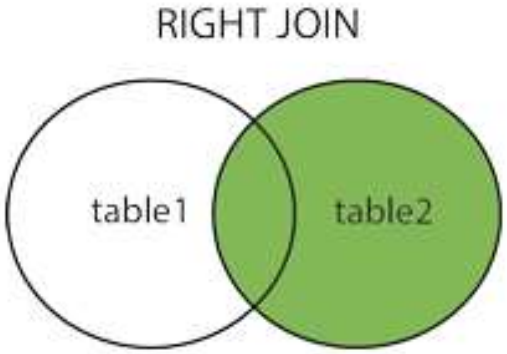
- Other types of join queries
  - (INNER) JOIN
    - Returns records that have matching values in both tables
  - LEFT JOIN
    - Returns all records from the left table, and the matched records from the right table
  - RIGHT JOIN
    - Returns all records from the right table, and the matched records from the left table
  - CROSS JOIN
    - Returns all possible combinations

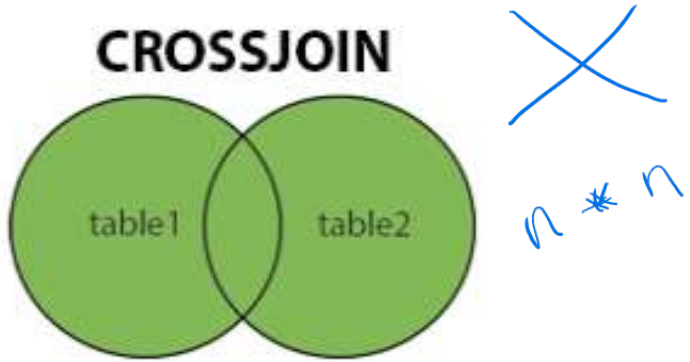
2. Intermediate DML

- Other types of join queries









2. Intermediate DML

- Other types of join queries
  - Examples in Korean

테이블A

A키	A컬럼
1	가
2	나

테이블B

B키	B컬럼
1	100
3	1500
4	20

INNER JOIN

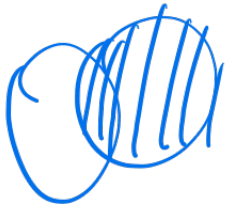
A키	A컬럼	B키	B컬럼
1	가	1	100

```
SELECT *
FROM 테이블A INNER JOIN 테이블B
ON 테이블A.A키 = 테이블B.B키
```

LEFT JOIN

A키	A컬럼	B키	B컬럼
1	가	1	100
2	나	NULL	NULL

```
SELECT *
FROM 테이블A LEFT JOIN 테이블B
ON 테이블A.A키 = 테이블B.B키
```



RIGHT JOIN

A키	A컬럼	B키	B컬럼
1	가	1	100
NULL	NULL	3	1500
NULL	NULL	4	20

```
SELECT *
FROM 테이블A RIGHT JOIN 테이블B
ON 테이블A.A키 = 테이블B.B키
```

3. Summary and Discussions

1. Intermediate DDL

unique, default, check + pk., fk ....

2. Intermediate DML
   - Join queries

common attribute

+

Join Condition

Select *

FROM employee, department

Join

WHERE (condition)