Project Report

Course Name: CSC3170-2022Fall
Team name: Database Messing System (DBMS)
Team Member:

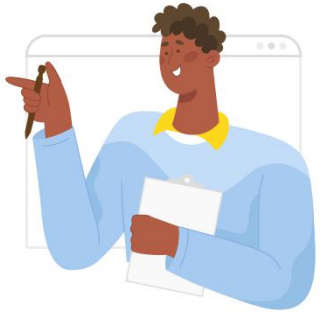
An Zihang	121090001
Hou Tianci	121090184
Shen Chihao	121020163
Tu yuzhao	121090519
Xia Yuyang	121090628
Zheng Yingqi	121090841

CONTENT

PART ONE

Structure

Project Structure
Database Structure



PART TWO

SQL Sentence

Basic sequence from UCB
Advance sequence

PART THREE

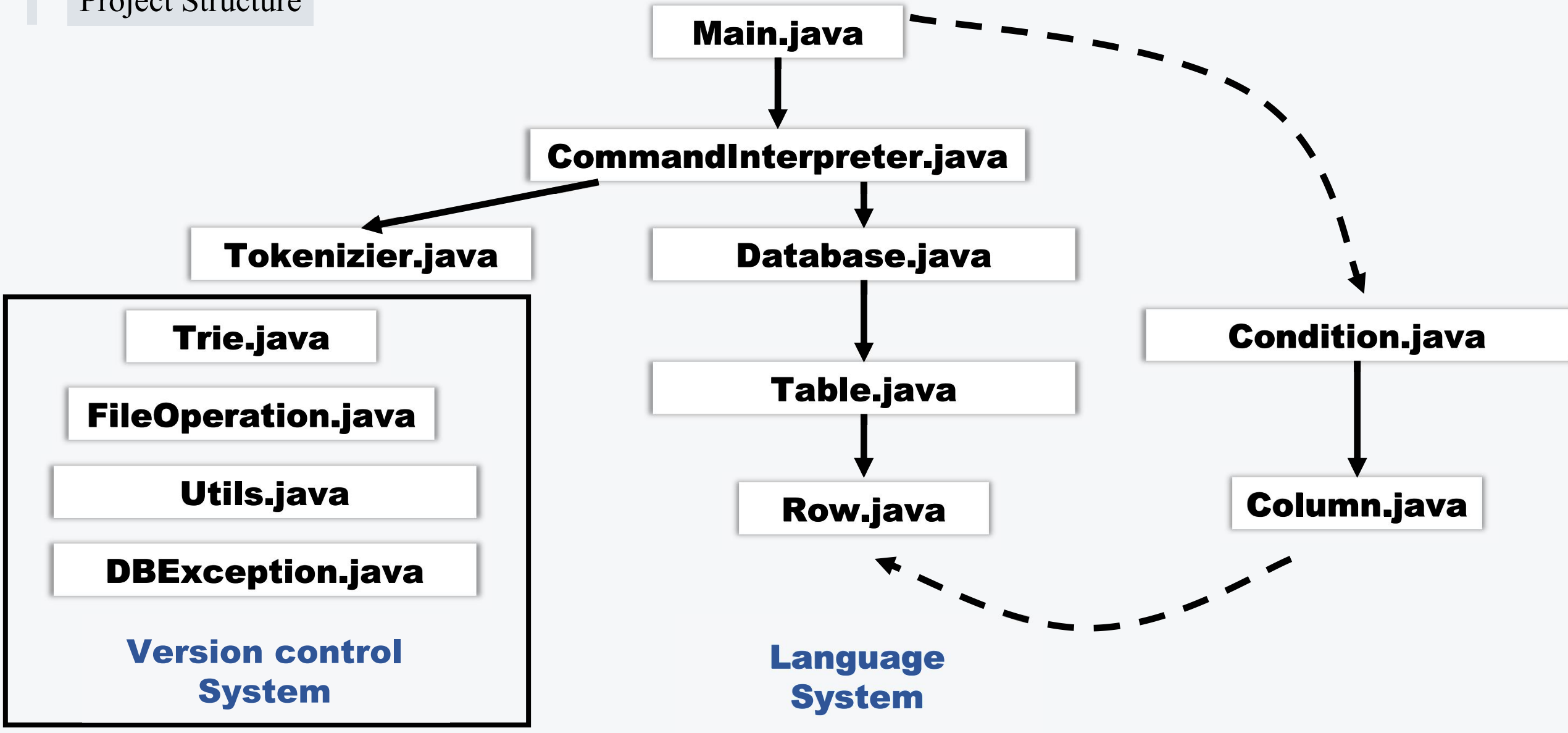
Version Control

PART FOUR

Application

Re-implement
Assignmnet 2

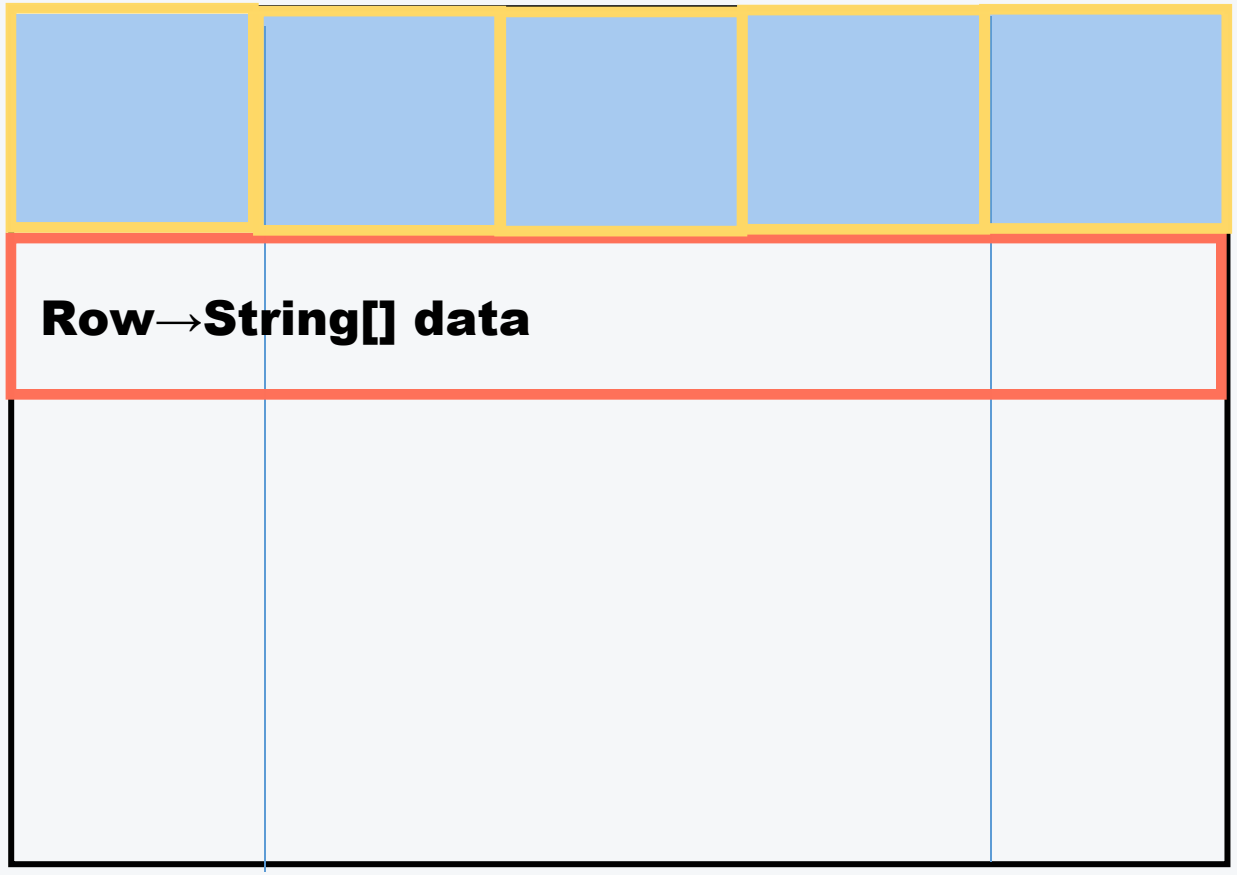
Project Structure



Database Structure

A table contains:
String[] _column_types;
String[] _column_titles;

Column Titles



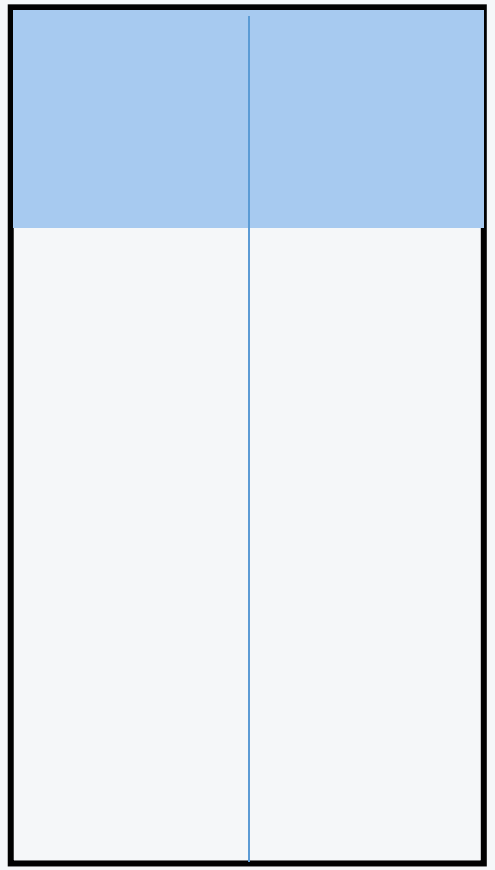


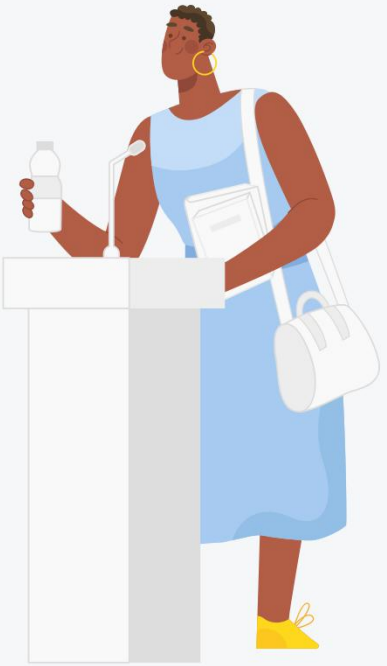
TABLE1→HashSet<Row>

TABLE2→HashSet<Row>

DATABASE→HashMap<string, table>

Basic grammar from UCB

- **<create statement>** ::= create table <name > <table definition > ;
- **<table definition>** ::= (<column name>$^{+}$,) | as <select clause>
- **<print statement>** ::= print <table name> ;
- **<insert statement>** ::= insert into <table name> values <literal>$^{+}$,;
- **<load statement>** ::= load <name> ;
- **<store statement>** ::= store <file name (no suffix)> <table name> ;
- **<exit statement>** ::= quit ; | exit ;
- **<select statement>** ::= <select clause>;
- **<select clause>** ::= select <column name>$^{+}$, from <tables> <condition clause>;
- **Operator in select clause**: =, <, <=, >, >=



Advanced grammar

About data type, we support:

3 data types: string, int and double

- Set data type for each column when creating table.
- Type validation when using functions.
  - Different data type supports different functions

About input, we support:

- Special symbol '*' representing all records.
- Rename the columns that are selected.

About output, we support:

- Organized output format of tables.
  - Format like MySQL.
- Clear exception message.

About integrity constraints, we support:

- Primary key.

```
> create table students(name string, stu_id int, score int);
> insert into students values('TYZ', '123456', '90');
> insert into students values('AZH', '234567', '91');
> insert into students values('HTC', '345678', '92');
```

```
> select * from students
...where score = '92';
Search results:
+----+------+-----+
|name|stu_id|score|
+----+------+-----+
| HTC|345678|   92|
+----+------+-----+
```

```
> select stu_id 'ID', score 'SCORE'
...from students where score = '92'
Search results:
+------+-----+
|    ID|SCORE|
+------+-----+
|345678|   92|
+------+-----+
```

```
> print students;
Table students:
+----+------+-----+
|name|stu_id|score|
+----+------+-----+
| HTC|345678|   92|
| AZH|234567|   91|
| XYY|123456|   91|
| ZYQ|123456|   89|
| TYZ|123456|   90|
+----+------+-----+
```

Advanced grammar

About functions, we support:

- **Aggregated functions:** avg, max, min, count, sum

  Syntax: select <function> <column name>+, from <table name>;

  Example:

```
> select avg score , max age, min age, count * from students, team;
Search results:
+-----------------+--------+--------+--------+
|       AVG(score)|MAX(age)|MIN(age)|COUNT(*)|
+-----------------+--------+--------+--------+
|62.333333333333336|    99.0|     3.0|       6|
+-----------------+--------+--------+--------+
```

- **Round function:**

  Syntax: select round <column name> <operator> <operand> reserve <number of bits reserved> from <table name>;

  Operator: plus, minus, times, divided_by

  Example:

```
> select round score times 3 reserve 3, stu_id from students;
Search results:
+-------------+----------+
|ROUND(score*3)|    stu_id|
+-------------+----------+
|          234|   XYYTXDY|
|          264| Aurora233|
|          174|  19260817|
|          180| TYZZZZZZZ|
|          255|1145141919|
|           15| 121090001|
+-------------+----------+
```

Advanced grammar

About Conditions, we support:

- in

  Syntax: select <column name>+, from <table name> where <column name> in <select clause>;

```
> select stu_id from students
...where stu_id in
...select * from students where gender = 'M';
Search results:
+----------+
|    stu_id|
+----------+
|   XYYTXDY|
|1145141919|
| 121090001|
+----------+
```

- order by

  Syntax: select <column name>+, from <table name> order by (asc/desc) <column name>+,;

```
> select * from students
...order by 'gender', 'score' desc;
Search results:
+----------+------+----+-----+
|    stu_id|gender|rank|score|
+----------+------+----+-----+
| Aurora233|     F|   A| 88.5|
| TYZZZZZZZ|     F|   A| 60.5|
|  19260817|     F|   A| 58.5|
|1145141919|     M|   A| 85.5|
|   XYYTXDY|     M|   A| 78.5|
| 121090001|     M|   A|  5.5|
+----------+------+----+-----+
```

Advanced grammar

About Conditions, we support:

- group by

  Syntax: select <column name>+, <function> <column name> from <table name>
  group by <column name>+,;

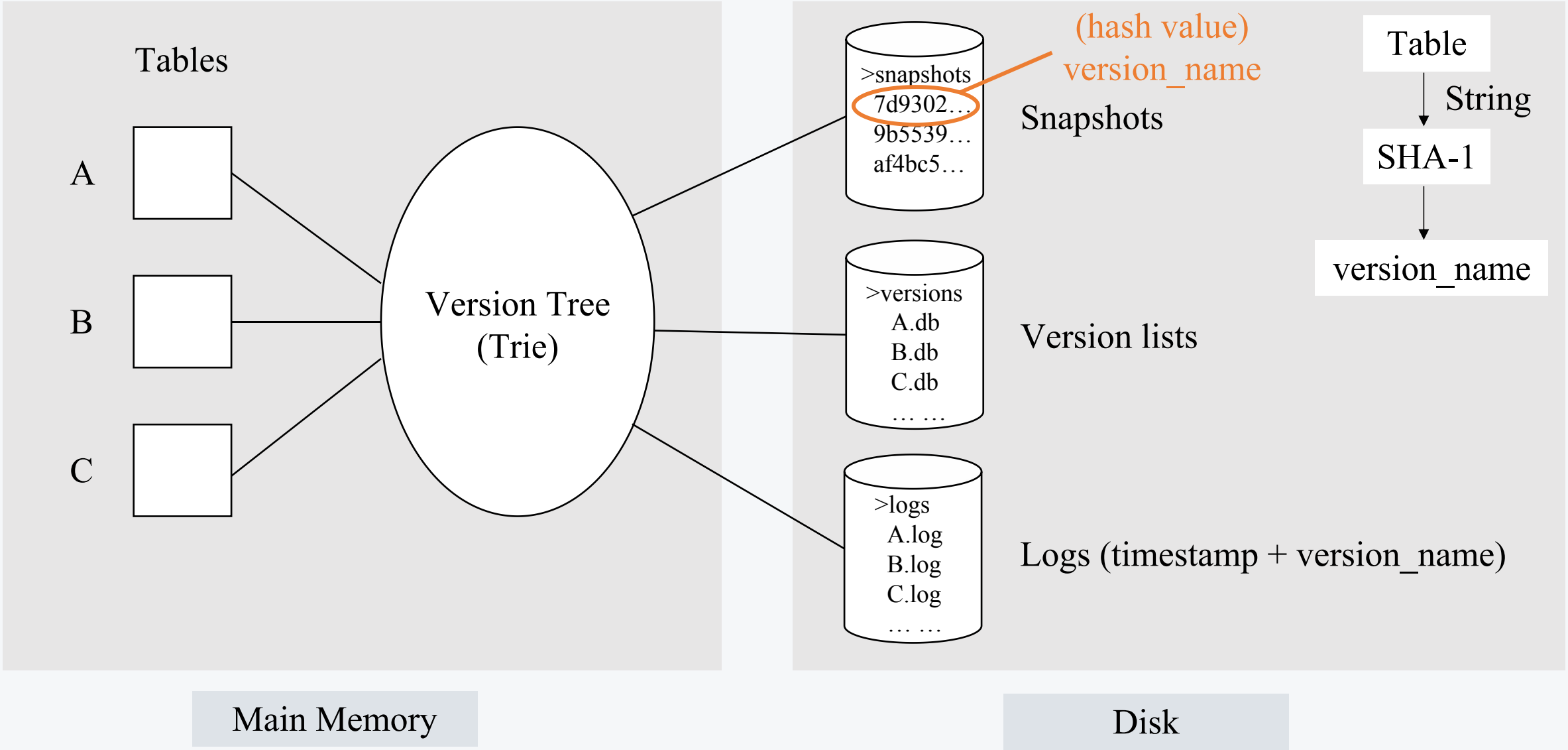
```
> select gender, count stu_id
...from students group by gender;
Search results:
+------+-------------+
|gender|COUNT(stu_id)|
+------+-------------+
|     M|            3|
|     F|            3|
+------+-------------+
```

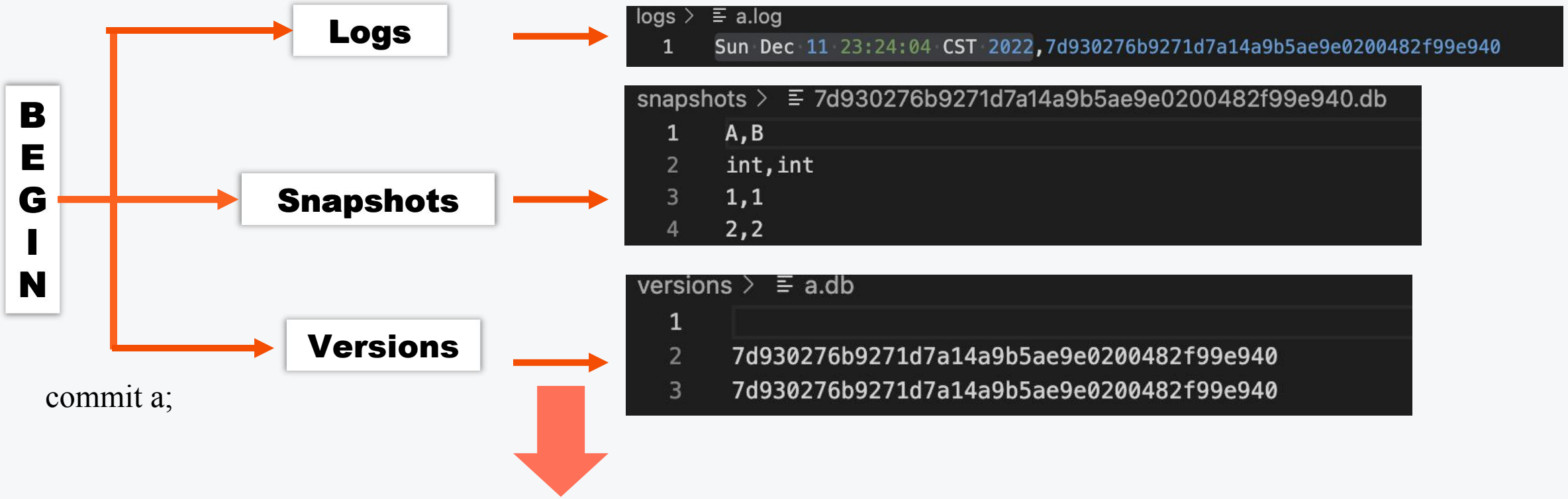
- between

  Syntax: select <column name>+, from <table name>
  where <column name> between <operand A> and <operand B>;

- like

  Syntax: select <column name>+, from <table name>
  where <column name> like <sample value>;
  supported operator: '_', '%'


Tables
A
B
C
Version Tree
(Trie)
>snapshots
7d9302…
9b5539…
af4bc5…
(hash value)
version_name
Snapshots
>versions
A.db
B.db
C.db
… …
Version lists
>logs
A.log
B.log
C.log
… …
Logs (timestamp + version_name)
Table
String
SHA-1
version_name
Main Memory
Disk



Syntax	Example: Table a (a.db)
<commit statement> ::= commit <table name>;	commit a;
<rollback to statement> ::= rollback <table name> to <literal>;	rollback a to '7d93';
<rollback at statement> ::= rollback <table name> at <literal>;	rollback a at 'Sun Dec 11 23:24:04 CST 2022';

```
================================
==   Start assignment2 test    ==
================================
Question 2:
  Passed.
============================
Question 3:
  Passed.
============================
Question 4:
  Passed.
============================
Question 5:
  Passed.
============================
Question 6:
  Passed.
============================
Question 7:
  Passed.
============================
Question 8:
  Passed.
============================
```

```
============================
Question 9:
  Passed.
============================
Question 10:
  Passed.
============================
Question 11:
  Passed.
============================
Question 12:
  Passed.
============================
Question 13:
  Passed.
============================
Question 14:
  Passed.
============================
Question 15:
  Passed.
============================
Question 16:
  Passed.
============================
```

Re-implement result

```
-- Create Table `regions` here

CREATE TABLE  regions(
    REGION_ID       INT(5),
    REGION_NAME     VARCHAR(25)     NOT NULL,
    PRIMARY KEY (REGION_ID)
);
```

```
> create table regions(
...region_id int, region_name string, primary key region_id);
> print regions;
Table regions:
+---------+-----------+
|region_id|region_name|
+---------+-----------+
+---------+-----------+
```

Question 1

```
SELECT FIRST_NAME AS "First Name", LAST_NAME AS "Last Name"
FROM employees
```

```
> select FIRST_NAME 'First Name', LAST_NAME 'Last Name'
...from employees;
Search results:
+----------+-----------+
|First Name|  Last Name|
+----------+-----------+
|   Randall|      Matos|
|      Jack| Livingston|
|    Diana|    Lorentz|
|.
 .
 .
+----------+--------+
```

Question 2

```
SELECT EMPLOYEE_ID, SALARY FROM employees ORDER BY SALARY
```

```
> select EMPLOYEE_ID, SALARY from employees
...order by 'SALARY';
Search results:
+-----------+--------+
|EMPLOYEE_ID|  SALARY|
+-----------+--------+
|        132| 2100.00|
|        136| 2200.00|
|        128| 2200.00|
|        127| 2400.00|
|        135| 2400.00|
 .
 .
 .
+-----------+--------+
```

Question 3

```
SELECT MAX(SALARY), MIN(SALARY) FROM employees
```

```
> select max SALARY, min SALARY from employee;
Search results:
+-----------+-----------+
|MAX(SALARY)|MIN(SALARY)|
+-----------+-----------+
|    24000.0|     2100.0|
+-----------+-----------+
```

Question 4

```
SELECT EMPLOYEE_ID, ROUND(SALARY/12, 2) FROM employees
```

```
> select round SALARY divided_by 12 reserve 2
...from employees;
Search results:
+-----------------+
|ROUND(SALARY/12~2)|
+-----------------+
|             2,000|
|           516.67|
|           291.67|
 .
 .
 .
+----------+-------+
```

Question 5

```
SELECT MANAGER_ID as 'EMPLOYEE_ID' FROM departments
WHERE MANAGER_ID NOT IN (SELECT employees.MANAGER_ID FROM employees);
```

```
> select MANAGER_ID 'EMPLOYEE ID' from departments
...where MANAGER_ID !='NULL'and MANAGER_ID not in
...select MANAGER_ID from employees;
Search results:
+-----------+
|EMPLOYEE ID|
+-----------+
|        200|
|        203|
|        204|
+-----------+
```

Question 6

```
SELECT EMPLOYEE_ID, PHONE_NUMBER FROM employees
WHERE DEPARTMENT_ID = 20 OR DEPARTMENT_ID = 100
```

```
> select EMPLOYEE_ID, PHONE_NUMBER from employees
...where DEPARTMENT_ID not in
...select DEPARTMENT_ID from employees
...where DEPARTMENT_ID!='20' and DEPARTMENT_ID !='100';
Search results:
+----------+------------+
|EMPLOYEE_ID|PHONE_NUMBER|
+----------+------------+
|        110|515.124.4269|
|        112|515.124.4469|
|        111|515.124.4369|
|        201|515.123.5555|
|        113|515.124.4567|
|        109|515.124.4169|
|        202|603.123.6666|
|        108|515.124.4569|
+----------+------------+
```

Question 7

```
SELECT FIRST_NAME FROM employees
WHERE FIRST_NAME LIKE '_a%'
```

```
> select FIRST_NAME from employee where FIRST_NAME like '_a%';
Search results:
+----------+
|FIRST_NAME|
+----------+
|     James|
|     Payam|
|    Daniel|
 .
 .
 .
+----------+
```

Question 8

```
SELECT JOB_ID, COUNT(*) FROM employees GROUP BY JOB_ID
```

```
> select JOB_ID, count EMPLOYEE_ID from employees, jobs
...group by JOB_ID;
Search results:
+----------+------------------+
|    JOB_ID|COUNT(EMPLOYEE_ID)|
+----------+------------------+
|  PU_CLERK|                 5|
|    ST_MAN|                 5|
|   IT_PROG|                 5|
|    MK_MAN|                 1|
 .
 .
 .
+----------+--------+
```

Question 9

```
SELECT DEPARTMENT_ID, AVG(SALARY), COUNT(*) FROM employees
GROUP BY DEPARTMENT_ID
HAVING COUNT(*)>10
```

```
> create table tmp as select DEPARTMENT_ID, count EMPLOYEE_ID 'NUMBER'
...from employees group by DEPARTMENT_ID;
> select DEPARTMENT_ID, avg SALARY from employees, tmp
...where NUMBER > '10' group by DEPARTMENT_ID;
Search results:
+-------------+-----------------+
|DEPARTMENT_ID|      AVG(SALARY)|
+-------------+-----------------+
|           80|8969.565217391304|
|           50|           3988.0|
+-------------+-----------------+
```

Question 10

```
SELECT FIRST_NAME, LAST_NAME FROM employees
WHERE MANAGER_ID IN(
    SELECT EMPLOYEE_ID FROM employees
    WHERE DEPARTMENT_ID IN (
        SELECT DEPARTMENT_ID FROM departments
        WHERE LOCATION_ID IN (
            SELECT LOCATION_ID FROM locations
            WHERE COUNTRY_ID = 'US'
        )
    )
)
```

```
> select FIRST_NAME, LAST_NAME from employees
...where MANAGER_ID in
...select EMPLOYEE_ID 'MANAGER_ID' from employees
...where DEPARTMENT_ID in
...select DEPARTMENT_ID from departments
...where LOCATION_ID in
...select LOCATION_ID from locations
...where COUNTRY_ID in
...select COUNTRY_ID from countries
...where COUNTRY_NAME = '"United States of America"';
Search results:
+----------+----------+
|FIRST_NAME| LAST_NAME|
+----------+----------+
|    Matthew|     Weiss|
|    Timothy|     Gates|
|    Randall|     Matos|
 .
 .
+----------+--------+
```

Question 11

```
SELECT EMPLOYEE_ID, SALARY FROM employees
WHERE  SALARY > ALL(
    SELECT AVG(SALARY) FROM employees
    GROUP BY DEPARTMENT_ID
    )
```

```
> create table avgs as
...select DEPARTMENT_ID, avg SALARY 'avg' from employees
...group by DEPARTMENT_ID;
> create table maxavg as
...select max avg 'maxAvg' from avgs;
> select EMPLOYEE_ID, SALARY from employees, maxavg
...where SALARY > maxAvg;
Search results:
+-----------+--------+
|EMPLOYEE_ID|  SALARY|
+-----------+--------+
|        100|24000.00|
+-----------+--------+
```

Question 12

```
SELECT DISTINCT EMPLOYEE_ID, SALARY
FROM employees e1
WHERE 4 = (SELECT COUNT(DISTINCT SALARY)
FROM employees  e2
WHERE e2.SALARY <= e1.SALARY);
```

```
> create table min1st as
...select min SALARY 'min1st' from employees;
> create table min2nd as
...select min SALARY 'min2nd' from employees, min1st
...where SALARY > min1st;
> create table min3rd as
...select min SALARY 'min3rd' from employees, min2nd
...where SALARY > min2nd;
> create table min4th as
...select min SALARY 'min4th' from employees, min3rd
...where SALARY > min3rd;
> select EMPLOYEE_ID, SALARY from employees, min4th
...where SALARY = min4th;
Search results:
+-----------+-------+
|EMPLOYEE_ID| SALARY|
+-----------+-------+
|        144|2500.00|
|        140|2500.00|
|        131|2500.00|
|        191|2500.00|
|        119|2500.00|
|        182|2500.00|
+-----------+-------+
```

Question 13

```
SELECT employees.EMPLOYEE_ID, employees.JOB_ID,
    employees.DEPARTMENT_ID, departments.DEPARTMENT_NAME
FROM employees JOIN departments ON (departments.DEPARTMENT_ID =
    employees.DEPARTMENT_ID)
WHERE departments.LOCATION_ID IN (
    SELECT LOCATION_ID FROM locations
    WHERE CITY = 'Seattle'
)
```

```
> create table tep as
...select DEPARTMENT_ID, DEPARTMENT_NAME from departments
...where DEPARTMENT_ID in
...select DEPARTMENT_ID from departments
...where LOCATION_ID in
...select LOCATION_ID from locations
...where CITY ='Seattle';
> select EMPLOYEE_ID, JOB_ID, DEPARTMENT_ID, DEPARTMENT_NAME
...from employees, tep;
Search results:
+-----------+----------+-------------+---------------+
|EMPLOYEE_ID|    JOB_ID|DEPARTMENT_ID|DEPARTMENT_NAME|
+-----------+----------+-------------+---------------+
|        102|     AD_VP|           90|      Executive|
|        119|  PU_CLERK|           30|     Purchasing|
|        116|  PU_CLERK|           30|     Purchasing|
|        114|    PU_MAN|           30|     Purchasing|
|        117|  PU_CLERK|           30|     Purchasing|
  .
  .
  .
 +-----------+--------+-------------+---------------+
```

Question 14

```
SELECT d.department_id `Department Name`, COUNT(*) `Number of Employees`
FROM departments d INNER JOIN employees e
ON d.department_id = e.department_id GROUP BY d.department_id
```

```
> select DEPARTMENT_ID 'Department Name', count EMPLOYEE_ID 'Number of Employees'
...from employees group by DEPARTMENT_ID;
Search results:
+---------------+------------------+
|Department Name|Number of Employees|
+---------------+------------------+
|            100|                 6|
|              0|                 1|
|             10|                 1|
  .
  .
 +-----------+--------+
```

Question 15

```
SELECT departments.DEPARTMENT_ID, departments.DEPARTMENT_NAME, employees.FIRST_NAME
FROM (departments JOIN employees ON (departments.MANAGER_ID = employees.EMPLOYEE_ID ))
```

```
> create table tep1 as
...select DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID from departments;
> create table tep2 as
...select EMPLOYEE_ID,FIRST_NAME from employees;
> create table preout as
...select DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, EMPLOYEE_ID,FIRST_NAME
...from tep1, tep2 where MANAGER_ID = EMPLOYEE_ID;
> select DEPARTMENT_ID, DEPARTMENT_NAME,FIRST_NAME from preout;
Search results:
+-------------+---------------+----------+
|DEPARTMENT_ID| DEPARTMENT_NAME|FIRST_NAME|
+-------------+---------------+----------+
|           90|      Executive|    Steven|
|           50|       Shipping|      Adam|
|           70|Public Relations|   Hermann|
  .
  .
 +-----------+--------+
```

Question 16

Thanks for your Attention!

Database Messing System (DBMS)
Dec 11th, 2022