Introduction to Structured Query Language

LAB 4

- **Writing Basic SQL SELECT Statements**

By Kanokwan Atchariyachanvanich

Faculty of Information Technology

KMITL

Database System Concepts

2/2565

OUTLINE ก่อนสอบกลางภาค

Date	SQL
9, 11 JAN 2023	• Lab Introduction • Introduction to DBLearn (SQL tool)
16, 18 JAN 2023	LAB 1 - Creating and Managing Tables (DDL)
23, 25 JAN 2023	LAB 2 - Including Constraints (DDL)
30 JAN, 1 FEB 2023	LAB 3 - Manipulating Data (DML)
6, 8 FEB 2023	LAB 4 - SQL SELECT Statements • Writing Basic
13, 15 FEB 2023	LAB 5 - SQL SELECT Statements • Restricting and Sorting Data
20, 22 FEB 2023	- ทวนก่อนสอบ Quiz 1
27 FEB, 1 MAR 2023	**Quiz 1: LAB 1 – LAB 5**

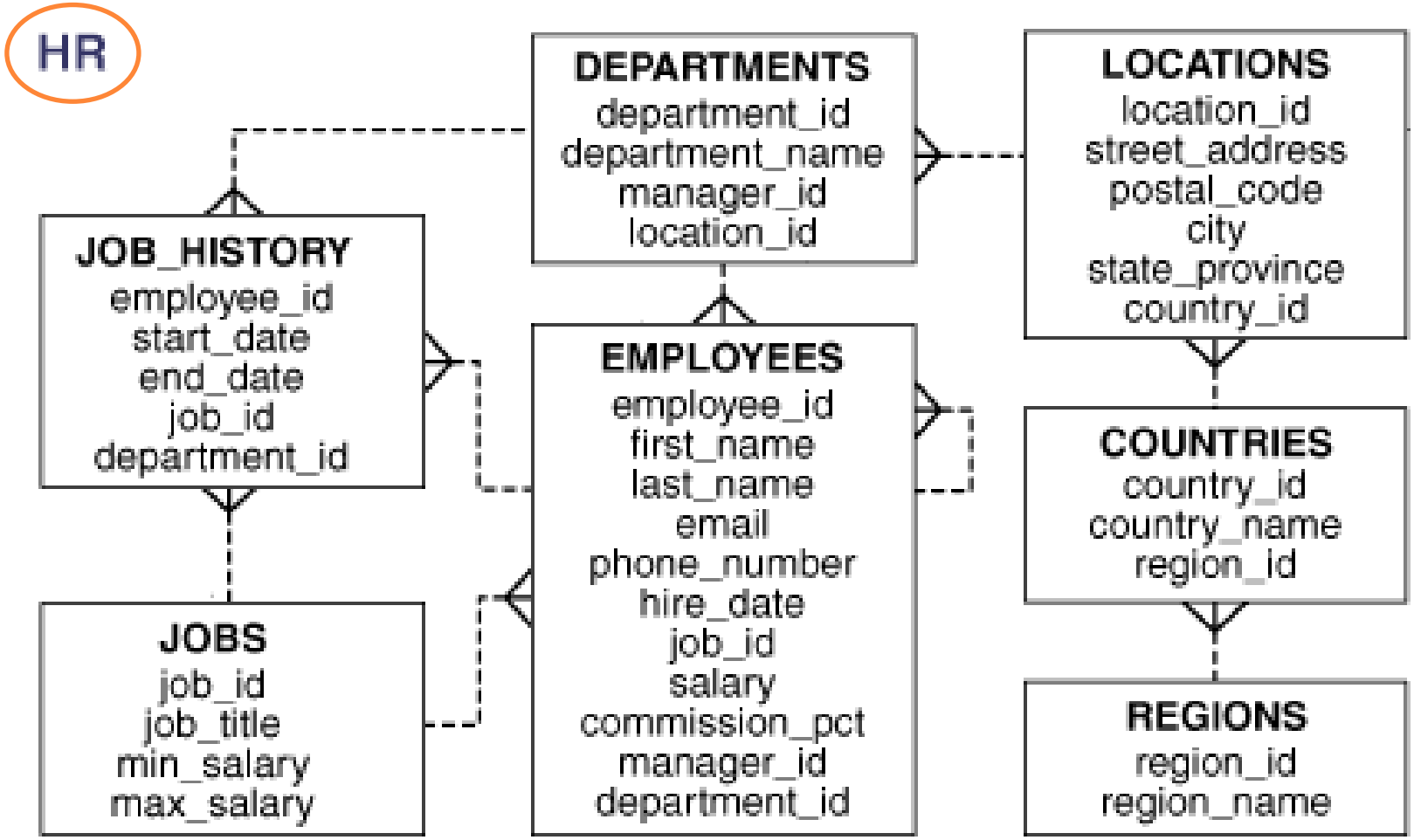
SQL STATEMENT

Type	SQL Statement
Data Manipulation Language (DML) ภาษาสำหรับจัดการข้อมูล คือส่วนของประโยค SQL ที่อนุญาตให้คุณควบคุมหรือจัดการข้อมูล	SELECT INSERT UPDATE DELETE MERGE CALL EXPLAIN PLAN LOCK TABLE
Data Definition Language (DDL) อธิบายส่วนของ SQL ที่อนุญาตให้สร้าง, เปลี่ยน, และทำลายอ็อบเจ็กต์ฐานข้อมูล อ็อบเจ็กต์ฐานข้อมูลเหล่านี้รวมถึงแบบแผน, ตาราง, มุมมอง ,ลำดับ, แคตาล็อก, ดัชนี, และ alias	CREATE ALTER DROP RENAME ANALYZE AUDIT COMMENT ASSOCIATE STATISTICS DISASSOCIATE STATISTICS FLASHBACK GRANT NOAUDIT PURGE REVOKE TRUNCATE UNDROP
Transaction Control จัดการ transaction จากการเปลี่ยนแปลงที่เกิดจาก DML	COMMIT ROLLBACK SAVEPOINT SET TRANSACTION

Objective

- Introduction to Structured query language
- Writing Basic SQL SELECT Statements
- Sorting Data

Tables Used in the Course



Example: EMPLOYEE Table Description

Column Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(20)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

How to know the column list

DESCRIBE departments;

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

Writing SQL Statements

- SQL statements are not case sensitive.
- SQL statements can be on one or more lines.
- Keywords cannot be abbreviated or split across lines.
- Keywords typically are entered in uppercase.
- Clauses are usually placed on separate lines.
- Indents are used to enhance readability.

Basic SELECT Statement

```
SELECT    *|{[DISTINCT] column_name | expression [alias],...}
FROM      table_name;
```

SELECT	**is a list of one or more columns**
*****	**selects all column**
DISTINCT	**suppresses duplicates**
Column_name|expression	**selects the named column or the expression**
alias	**gives selected column different headings**
FROM *table_name*	**specifies the table containing the columns**

SELECTING ALL COLUMNS

```
SELECT  *
FROM    departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700
120	Treasury		1700
130	Corporate Tax		1700
140	Control And Credit		1700
150	Shareholder Services		1700
160	Benefits		1700
170	Manufacturing		1700
180	Construction		1700
190	Contracting		1700
200	Operations		1700

....

27 rows selected.

SELECTING SPECIFIC COLUMNS

ชื่อคอลัมน์มาจากคอลัมน์ในตาราง
หลังคำ FROM

```
SELECT  department_id, location_id
FROM    departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
30	1700
40	2400
50	1500
60	1400
70	2700
80	2500
90	1700
100	1700
110	1700
120	1700
130	1700
140	1700
150	1700
160	1700
170	1700
180	1700
190	1700
200	1700

....

27 rows selected.

Arithmetic Expressions

- Create expressions by using arithmetic operators.
- Arithmetic expression can contain column names, constant numeric values, and the arithmetic operators.

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide

Using Arithmetic Operators

```
SELECT    last_name, salary, salary + 300
FROM      employees;
```

LAST_NAME	SALARY		SALARY+300
King	24000		24300
Kochhar	17000		17300
De Haan	17000		17300
Hunold	9000		9300
Ernst	6000		6300
Austin	4800		5100
Pataballa	4800		5100
Lorentz	4200		4500
Greenberg	12000		12300
Faviet	9000		9300
Chen	8200		8500
Sciarra	7700		8000
....			
Hartstein	13000		13300
Fay	6000		6300
Mavris	6500		6800
Baer	10000		10300
Higgins	12000		12300
Gietz	8300		8600

107 rows selected.

Operator Precedence

* / + –

- Multiplication and division take priority over addition and subtraction.
- Operators of the <u>same priority</u> are evaluated from <u>left to right.</u>
- <u>Parentheses are used to force prioritized</u> evaluation and to clarify statements.

Operator Precedence (cont.)

```
SELECT    last_name, salary, 12*salary+100
FROM      employees;
```

LAST_NAME	SALARY		12*SALARY+100
King	24000		288100
Kochhar	17000		204100
De Haan	17000		204100
Hunold	9000		108100
Ernst	6000		72100
Austin	4800		57700
Pataballa	4800		57700
Lorentz	4200		50500
Greenberg	12000		144100
Faviet	9000		108100
....			
Mavris	6500		78100
Baer	10000		120100
Higgins	12000		144100
Gietz	8300		99700

107 rows selected.

Using Parentheses

```
SELECT    last_name, salary, 12* (salary+100)
FROM      employees;
```

LAST_NAME	SALARY		12*(SALARY+100)
King	24000		289200
Kochhar	17000		205200
De Haan	17000		205200
Hunold	9000		109200
Ernst	6000		73200
Austin	4800		58800
Pataballa	4800		58800
Lorentz	4200		51600
Greenberg	12000		145200
Faviet	9000		109200
Chen	8200		99600
Sciarra	7700		93600
Urman	7800		94800
Popp	6900		84000
....			
Fay	6000		73200
Mavris	6500		79200
Baer	10000		121200
Higgins	12000		145200
Gietz	8300		100800

107 rows selected.

Defining a Null Value

- A null is a value that is unavailable, unassigned, unknown, or inapplicable.
- A null is not the same as zero or a blank space.

```
SELECT last_name, job_id, salary, commission_pct
FROM   employees;
```

Null

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	
....			
Kochhar	AD_VP	17000	
....			
Zlotkey	SA_MAN	10500	.2
Abel	SA_REP	11000	.3
Taylor	SA_REP	8600	.2
....			
Gietz	AC_ACCOUNT	8300	

107 rows selected.

Null

Remark: NOT NULL and PRIMARY KEY prevents nulls in column

Null Values in Arithmetic Expressions

- **Arithmetic expressions containing a null value evaluate to null.**

```
SELECT last_name, 12* salary*commission_pct
FROM     employees;
```

Null

LAST_NAME	12*SALARY*COMMISSION_PCT
King	
Kochhar	
....	
Zlotkey	25200
Abel	39600
Taylor	20640
....	
Gietz	

107 rows selected.

Null

Defining a Column Alias (นามแฝง)

- **Oracle**
- การใช้ Alias แทนชื่อคอลัมน์เดิมในตาราง
- Default แสดงตัวพิมพ์ใหญ่
- ชื่อattribute AS นามแฝง
- ชื่อattribute เว้นวรรค นามแฝง
- นามแฝงต้องมี double quotation ("__") ถ้านามแฝงนั้นมีช่องว่าง ตัวอักษรพิเศษ หรือ เป็น case sensitive

- **MySQL**
- การใช้ Alias แทนชื่อคอลัมน์เดิมในตาราง
- รูปแบบการเขียนคือ
  - ชื่อattribute AS นามแฝง
  - ชื่อattribute เว้นวรรค นามแฝง
- ถ้านามแฝงนั้นมีช่องว่าง ตัวอักษรพิเศษ นามแฝงต้องมี back-tick `` หรือ double quote " " หรือ single quote ' '
- ต้องมี back tick(`_`) สำหรับอ้างอิงไปยังคอลัมน์ที่ตั้งนามแฝงไว้

Using Column Aliases (MySQL)

```
SELECT   last_name AS name, commission_pct comm
FROM     employees;
```

name	comm
King	
Kochhar	
De Haan	

....

107 rows selected.

```
SELECT   last_name Name, salary*12 `Annual Salary`
FROM     employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

....

107 rows selected.

Concatenation Operator

- MySql uses Concat function
  - concat('string1', 'string2', 'string3',….)
  - concat(ชื่อattribute1, 'string2', ชื่อattribute2,….)
  - Example: concat(firstname, ' _ ' , lastname)
- Oracle A concatenation operator:
  - Concatenates columns or character strings to other columns

    เชื่อมคอลัมน์หรือตัวอักษรกับคอลัมน์อื่น
  - Is represented by two vertical bars (||)
  - Creates a resultant column that is a character expression

Using the Concatenation Operator (MySQL)

```
SELECT    concat(last_name,  job_id) Employees
FROM      employees;
```

Employees
KingAD_PRES
KochharAD_VP
De HaanAD_VP
HunoldIT_PROG
ErnstIT_PROG
LorentzIT_PROG
MourgosST_MAN
RajsST_CLERK

....

107 rows selected.

Literal Character Strings

- กำหนด ตัวอักษร หรือ วันที่ ลงในบรรทัด SELECT
- A literal is a character, or a date included in the SELECT list.
- Date and character literal values must be enclosed within single quotation ('_') marks or double quotation (" _ ") marks.
- Each character string is output once for each row returned.
- In DBLearning (MySQL), Date format -> YYYY-MM-DD

Using Literal Character Strings (MySQL)

```
SELECT    concat(last_name, ' is a ', job_id) ` Employee Details `
FROM      employees;
```

Employee Details
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG
Lorentz is a IT_PROG
Mourgos is a ST_MAN
Rajs is a ST_CLERK

....

107 rows selected.

Using Literal Character Strings (ORACLE)

```
SELECT    last_name || ' is a '  || job_id  AS "Employee Details"
FROM      employees;
```

Employee Details
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG
Lorentz is a IT_PROG
Mourgos is a ST_MAN
Rajs is a ST_CLERK

....

107 rows selected.

Duplicate Rows

```
SELECT    department_id
FROM      employees;
```

DEPARTMENT_ID
90
90
90
60
60
60
50
50
50

....

107 rows selected.

Eliminating Duplicate Rows

- Eliminate duplicate rows by using the DISTINCT keyword in the SELECT clause.

```
SELECT   DISTINCT department_id
FROM     employees;
```

DEPARTMENT_ID
10
20
30
40
50
60
70
80
90
100
110

12 rows selected.

ORDER BY CLAUSE

- Sort rows with the ORDER BY clause
- The ORDER BY clause comes last in the SELECT statement.

```
SELECT      last_name, job_id, department_id, hire_date
FROM        employees
ORDER  BY  hire_date ;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
King	AD_PRES	90	17-JUN-87
Whalen	AD_ASST	10	17-SEP-87
Kochhar	AD_VP	90	21-SEP-89
Hunold	IT_PROG	60	03-JAN-90
Ernst	IT_PROG	60	21-MAY-91

....

107 rows selected.

SUMMARY

- **In this lesson, you should have learned how to:**
- **Retrieve data from 1 table by specifying column name or expression**

```
SELECT    *|{[DISTINCT]  column_name | expression [alias],...}
FROM      table_name
```

Sorting Data

- เรียงลำดับแถวข้อมูลของผลลัพธ์จาก Query ด้วย ORDER BY
- 2 รูปแบบ
  - ASC (Ascending Order) หรือไม่ระบุ (by default) เรียงจาก น้อยไปมาก, A-Z, จากวันที่อดีตมาวันที่ปัจจุบัน
  - DESC (Descending Order) เรียงจาก มากไปน้อย, Z-A, จากวันที่ปัจจุบัน
  - ย้อนไปวันที่อดีต

```
SELECT    *|{[DISTINCT]  column_name | expression [alias],...}
FROM      table_name
[ ORDER BY {column_name, expr}  [ASC | DESC] ] ;
```

Sorting in Ascending Order

```
SELECT      last_name, job_id, department_id, hire_date
FROM        employees
ORDER BY    hire_date ASC ;
```

last_name	job_id	department_id	hire_date
King	AD_PRES	90	1987-06-17
Whalen	AD_ASST	10	1987-09-17
Kochhar	AD_VP	90	1989-09-21
Hunold	IT_PROG	60	1990-01-03
Ernst	IT_PROG	60	1991-05-21
De Haan	AD_VP	90	1993-01-13
Mavris	HR_REP	40	1994-06-07
Baer	PR_REP	70	1994-06-07
Higgins	AC_MGR	110	1994-06-07
Gietz	AC_ACCOUNT	110	1994-06-07

Sorting in Descending Order

```
SELECT      last_name, job_id, department_id, hire_date
FROM        employees
ORDER BY    hire_date DESC ;
```

last_name	job_id	department_id	hire_date
Banda	SA_REP	80	2000-04-21
Kumar	SA_REP	80	2000-04-21
Ande	SA_REP	80	2000-03-24
Markle	ST_CLERK	50	2000-03-08
Lee	SA_REP	80	2000-02-23
Philtanker	ST_CLERK	50	2000-02-06
Geoni	SH_CLERK	50	2000-02-03
Zlotkey	SA_MAN	80	2000-01-29
Marvins	SA_REP	80	2000-01-24
Grant	SH_CLERK	50	2000-01-13

....

107 rows selected.

Sorting by Column Alias

```
SELECT      employee_id, last_name, salary*12 annsal
FROM        employees
ORDER BY    annsal ;
```

EMPLOYEE_ID	LAST_NAME	ANNSAL
132	Olson	25200
128	Markle	26400
136	Philtanker	26400
127	Landry	28800
135	Gee	28800
119	Colmenares	30000
140	Patel	30000
144	Vargas	30000
191	Perkins	30000
182	Sullivan	30000
131	Marlow	30000
118	Himuro	31200
143	Matos	31200

....

107 rows selected.

Sorting by Column Alias with Space (MySQL)

```
SELECT      employee_id, last_name, salary*12 `annual sal`
FROM        employees
ORDER BY    `annual sal` ;
```

EMPLOYEE_ID	LAST_NAME	ANNSAL
132	Olson	25200
128	Markle	26400
136	Philtanker	26400
127	Landry	28800
135	Gee	28800
119	Colmenares	30000
140	Patel	30000
144	Vargas	30000
191	Perkins	30000
182	Sullivan	30000
131	Marlow	30000
118	Himuro	31200
143	Matos	31200

....

107 rows selected.

Sorting by Multiple Columns

- The order of ORDER BY list is the order of sort.
- Example: Display the last name and salaries of all employees. Order the results by department number, and then in descending order by salary

```
SELECT      last_name, department_id, salary
FROM        employees
ORDER BY    department_id, salary DESC ;
```

LAST_NAME	DEPARTMENT_ID	SALARY
Whalen	10	4400
Hartstein	20	13000
Fay	20	6000
Raphaely	30	11000
Khoo	30	3100
Baida	30	2900
Tobias	30	2800
Himuro	30	2600
Colmenares	30	2500
Mavris	40	6500
Fripp	50	8200
Weiss	50	8000
Kaufling	50	7900

....

107 rows selected.

Remark: You can sort by a column that is not in the SELECT list.

ORDER BY CLAUSE

- **Sort rows with the ORDER BY clause**
  - **ASC: ascending order, default**
  - **DESC: descending order**
- **The ORDER BY clause comes last in the SELECT statement.**

```
SELECT    *|{[DISTINCT]  column_name | expression [alias],...}
FROM      table_name
[ WHERE   condition(s) ]
[ ORDER BY {column_name, expr}  [ASC | DESC] ] ;
```