

Class 1 [DBMS , SQL]

▼ DBMS?

데이터베이스란 데이터를 저장, 검색, 관리하고 조직화하는데 사용하는 컴퓨터 시스템
DBMS는 DataBase Management System의 약자로 사용자나 프로그램이 데이터에 접근할 수 있게 해줍니다.

▼ 데이터베이스 유형

• Relational Database 관계형 데이터베이스

테이블 (형과 열로 이루어진) 형태로 데이터를 저장하며, SQL(Structured Query Language)을 사용해 데이터에 접근합니다. 이러한 데이터베이스는 일반적으로 ACID(Atomicity, Consistency, Isolation, Durability)속성을 유지합니다.

대표적 관계형 데이터베이스는

MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server등이 있습니다.

• No SQL (Not Only SQL) 데이터베이스

관계형 데이터베이스와는 달리 여러 유형의 데이터 구조를 허용하며, 테이블 간의 관계를 구성하지 않습니다.

대용량의 분산 데이터를 처리하는데 유용하며, 확장성이 높습니다.

No SQL의 네가지 유형

Document → 문서형

Key - Value → 키-값

Column-Family → 칼럼-패밀리

Graph → 그래프

대표적인 NoSQL 데이터베이스

MongoDB, Redis, Cassandra, Neo4j

기타 데이터베이스

Hierarchical - 계층형 데이터베이스

Network - 망형

Object-Oriented - 객체지향형

Objenct-Relational - 객체관계형

▼ 데이터베이스 구성 요소

- **Schema 스키마**

데이터베이스의 구조와 제약조건을 정의

테이블, 인덱스, 뷰, 프로시저 등의 객체를 포함

- **Table 테이블**

행과 열로 이루어진 데이터 저장 구조

각 열은 데이터의 속성을, 각 행은 데이터의 레코드를 나타냄

- **Index 인덱스**

데이터 검색 속도를 높이기 위해 사용되는 데이터 구조

일종의 데이터베이스 '색인'으로, 특정 열의 값에 따라 레코드를 빠르게 찾을 수 있게 함

- **View 뷰**

데이터베이스의 가상 테이블

실제로 데이터를 저장하지 않지만 기본 테이블로부터 데이터를 참조하고 질의 수행

데이터 접근을 단순화하거나, 보안 목적으로 특정 데이터에 대한 접근을 제한하는 데 사용

- **Transaction 트랜잭션**

하나의 논리적 작업 단위

데이터의 일관성과 동시성을 유지하기 위해 ACID속성을 따름

- **Stored Procedure 저장 프로시저**

데이터베이스 서버에서 실행되는 일련의 SQL명령어로 구성된 코드 블록

코드 재사용, 성능 향상 및 보안 향상과 같은 이점 제공

- **Trigger 트리거**

특정 이벤트(INSERT, UPDATE, DELETE 등)가 발생할 때 자동으로 실행되는 사용자 정의 코드

데이터의 일관성을 유지하거나, 추가 로직을 적용하는 데 사용

- **Constraint 제약 조건**

데이터의 무결성을 유지하기 위해 테이블에 적용되는 규칙

NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK 등이 있음

- **DBMS 데이터베이스 관리 시스템**

데이터베이스를 생성하고, 조작하며, 관리하는 소프트웨어

사용자와 프로그램이 데이터베이스와 상호작용 가능하게 함

관계형 DBMS와 No SQL DBMS로 나뉨

▼ DBMS 종류

DBMS	제작사	작동 운영체제	최신 버전	기타
MySQL	Oracle	Unix, Linux, Windows, Mac	8.0	오픈 소스(무료), 상용
MariaDB	MariaDB	Unix, Linux, Windows	10.6	오픈 소스(무료), MySQL 초기 개발자들이 독립해서 만들
PostgreSQL	PostgreSQL	Unix, Linux, Windows, Mac	12	오픈 소스(무료)
Oracle	Oracle	Unix, Linux, Windows	18c	상용 시장 점유율 1위
SQL Server	Microsoft	Windows	2019	주로 중/대형급 시장에서 사용
DB2	IBM	Unix, Linux, Windows	11	메인프레임 시장 점유율 1위
Access	Microsoft	Windows	2019	PC용
SQLite	SQLite	Android, iOS	3	모바일 전용, 오픈 소스(무료)

▼ SQL?

Structured Query Language

관계형 데이터베이스 관리 시스템(RDBMS)에서 데이터를 검색, 수정, 추가 및 삭제하기 위해 사용되는 표준 프로그래밍 언어

관계형 데이터베이스에서 데이터를 조작하고 관리하기 위한 명령어와 구문을 제공

여러가지 RDBMS

- Oracle, MySQL, PostgreSQL, Microsoft SQL Server 등

ANSI(미국 국립 표준 협회) 및 ISO(국제 표준화 기구)에 의해 표준화

▼ SQL 역할

- 데이터 검색(조회)

SELECT 문을 사용하여 데이터베이스의 특정 테이블에서 원하는 데이터를 검색하거나 조회

필터링, 정렬, 그룹화, 결합 등 다양한 결과 출력 가능

- 데이터 삽입

INSERT 문을 사용하여 테이블에 새로운 레코드(행)를 추가 가능

각 열에 대한 값을 지정하거나 다른 테이블로부터 값을 복사 가능

- 데이터 수정

UPDATE 문을 사용하여 테이블의 기존 레코드(행)을 수정 가능

조건을 지정하여 특정 레코드만 수정 가능

- 데이터 삭제

DELETE 문을 사용하여 테이블에서 레코드(행)을 삭제 가능

데이터베이스, 테이블, 인덱스, 뷰 등의 구조를 생성, 수정, 삭제 가능

- 테이블 및 데이터베이스 구조 조작

CREATE, ALTER, DROP 문을 사용하여 데이터베이스, 테이블, 인덱스, 뷰 등의 구조를

생성, 수정, 삭제 가능

- 트랜잭션 제어

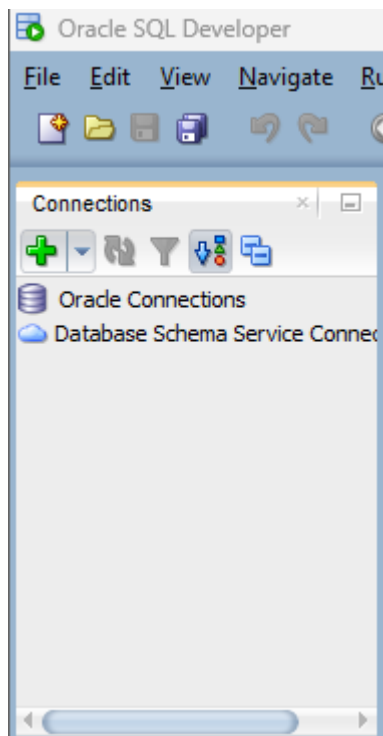
트랜잭션은 데이터베이스의 논리적 작업 단위로, **COMMIT, ROLLBACK, SAVEPOINT** 등의 명령어를 사용하여 트랜잭션을 제어하고 데이터의 일관성 유지

- 데이터베이스 접근 제어

GRANT, REVOKE 명령어를 사용하여 사용자에게 데이터베이스 객체에 대한 권한 부여 및 취소

데이터의 보안 유지

▼ Oracle



+ 버튼 눌러서 만듦

Connection Name	Connection Details
orade	SYSTEM@//localh...

Name: orade

Database Type: Oracle

User Info Proxy User

Authentication Type: Default

Username: SYSTEM Role: default

Password: **** ☒ Save Password

Connection Type: Basic

Details Advanced

Hostname: localhost

Port: 1521

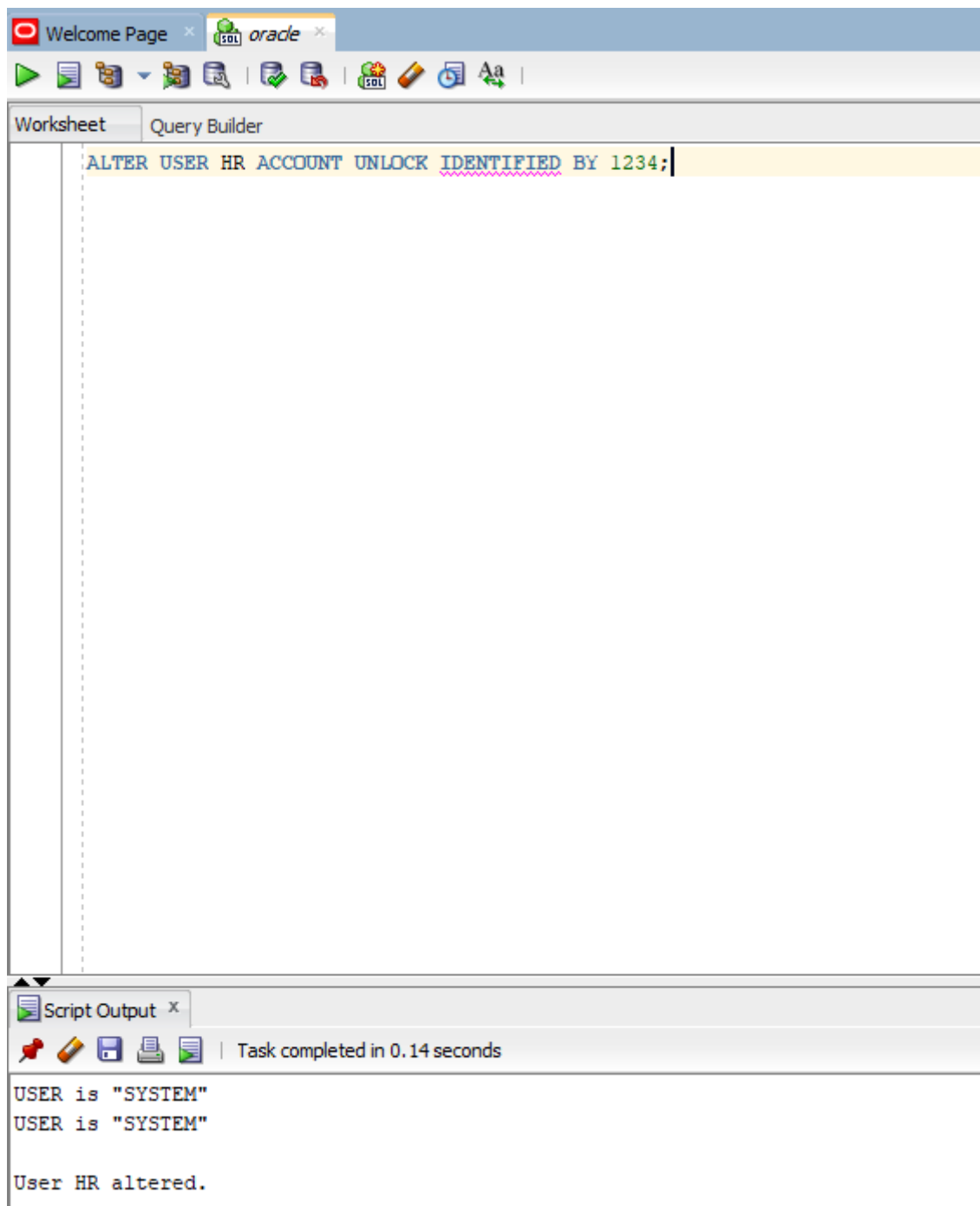
☒ SID: xe

☐ Service name:

Status: Success

Buttons: Help, Save, Clear, Test, Connect, Cancel

Test 후 왼쪽 아래 Success 나오면 성공 / 비번 0000



학습용 유저 만들기

New / Select Database Connection

Connection Name	Connection Details
orade	SYSTEM@//localh...

Name: VMHR

Database Type: Oracle

User Info: Proxy User

Authentication Type: Default

Username: HR

Password: ****

Role: default

Save Password: ☒

Connection Type: Basic

Details: Advanced

Hostname: localhost

Port: 1521

SID: ☒ xe

Service name: ☐

Status: Success

Buttons: Help, Save, Clear, Test, Connect, Cancel

비번 1234

Welcome Page x orade x VMHR x

Worksheet Query Builder

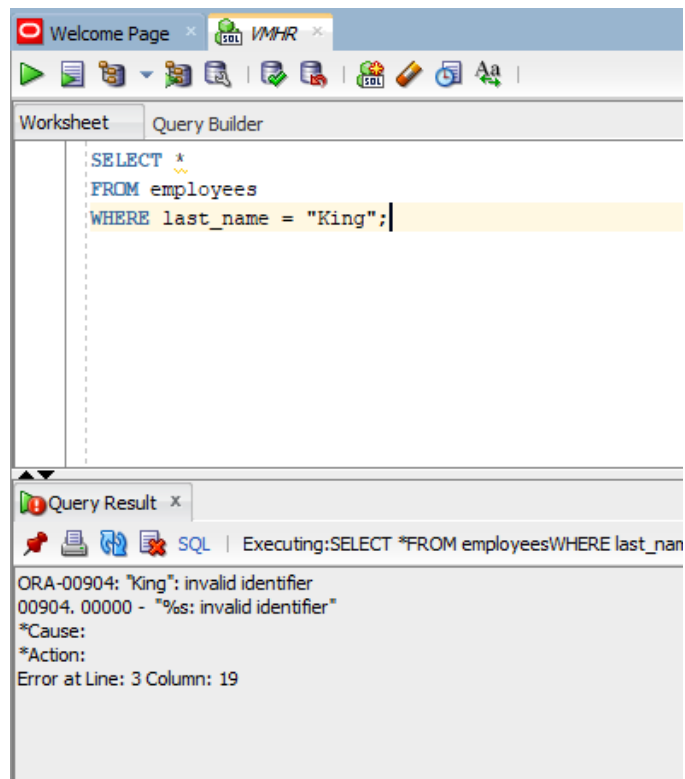
SELECT * FROM tab;

Query Result x

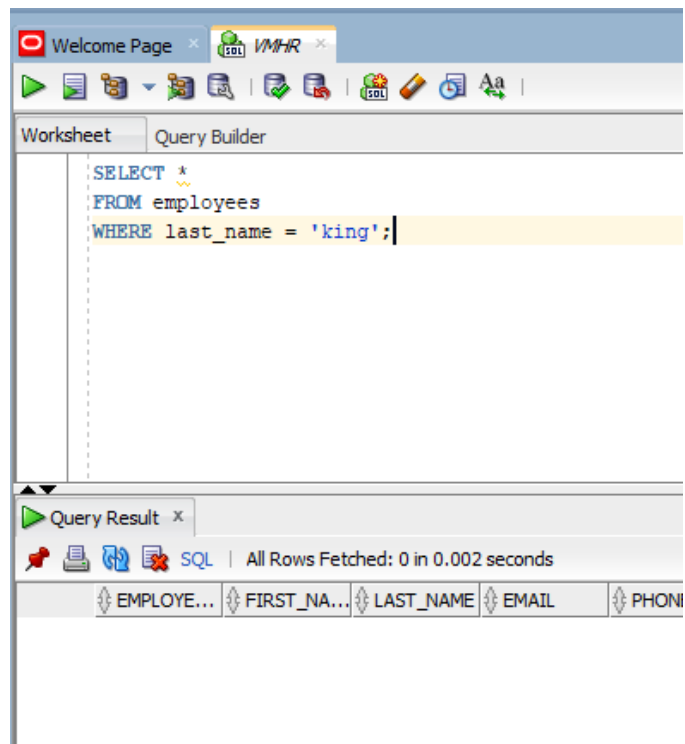
SQL | All Rows Fetched: 8 in 0.029 seconds

	TNAME	TABTYPE	CLUSTERID
1	COUNTRIES	TABLE	(null)
2	DEPARTMENTS	TABLE	(null)
3	EMPLOYEES	TABLE	(null)
4	EMP_DETAILS_VIEW	VIEW	(null)
5	JOBS	TABLE	(null)
6	JOB_HISTORY	TABLE	(null)
7	LOCATIONS	TABLE	(null)
8	REGIONS	TABLE	(null)

WHERE절 오류

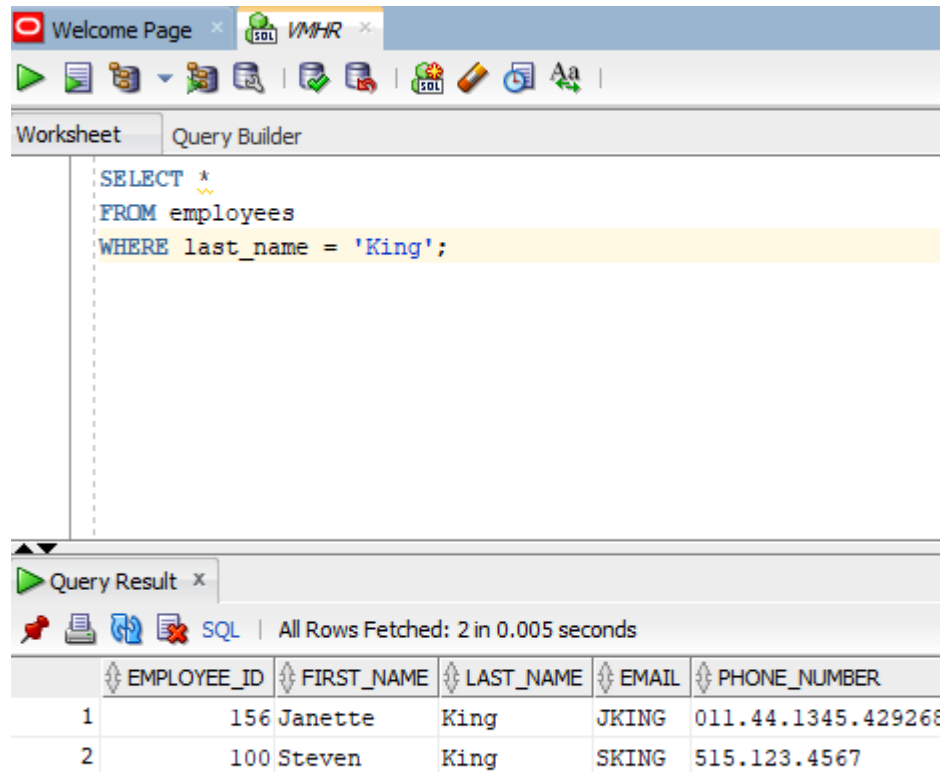


“ “ 큰따옴표 오류



소문자 오류 * 값의 경우 소문자 사용 X

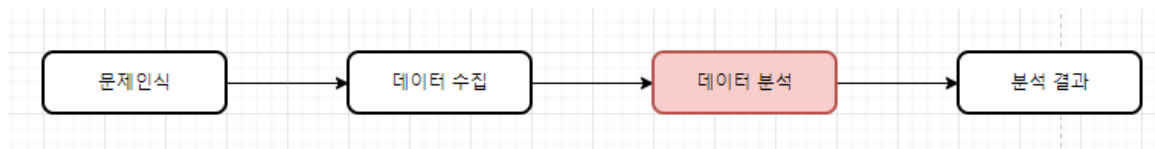
정상작동



▼ 데이터 분석 과정

데이터 분석과정

문제인식 → 데이터수집 → 데이터 분석 → 분석결과



전처리 (pre - processing)

데이터를 수집하고 가공하는 과정을 특히 **전처리**라고 하며 데이터 전체 분석 과정의 약 80%를 차지한다. / 분석은 컴퓨터가 다 해줌 /

정형 데이터와 비정형 데이터

- 정형 데이터 : 틀이 잡힌 데이터 (ex 엑셀 등등)
- 비정형 데이터 : 틀이 없는 다양하고 방대한 데이터(ex 트위터, 음악, 그림 등등)

▼ SELECT - 데이터 조회

가장 많이 사용되는 구문

기본형

SELECT 열이름 **FROM** 테이블 이름;

SELECT 열이름1, 열이름2

FROM 테이블이름;

SELECT 열이름1, 열이름2

FROM 테이블이름;

주의사항

1. SQL구문은 **대문자**로 사용하자(관용적인 표현, 소문자도 가능) / 나머지는 소문자
2. 줄 바꿈은 특별히 제한하는 것이 없다.
3. 마지막에 ‘;’ **세미콜론**으로 마쳐야만 한다.

```
1 SELECT first_name, last_name FROM hr.employees;
```

Hermann	Baer
Shelli	Baida
Amit	Banda
Elizabeth	Bates
Sarah	Bell
David	Bernstein
Laura	Bissot
Harrison	Bloom
Alexis	Bull

구문은 대문자, 나머지는 소문자

```
1 v SELECT first_name, last_name
```

```
2 FROM hr.employees;
```

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin

줄바꿈은 제한 없다

```
1 v SELECT first_name, last_name
```

```
2 FROM hr.employees;
```

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin

SELECT * = 전체조회

```

1 SELECT *
2 FROM hr.employees;
3

```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG	9000	-	102	60

웹을 사용할 때에는 hr.employees;

프로그램으로 실행 할 때에는 그냥 employees;

ctrl + space 누르면 자동완성 가능

▼ ORDER BY - 정렬 조회

SELECT*

FROM 테이블 이름

ORDER BY 열이름 **ASC or DESC**

ASC 오름차순 - 예전 게시글이 먼저

DESC 내림차순 - 최근이 먼저

```

1  SELECT *
2  FROM hr.employees
3  ORDER BY employee_id DESC;

```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
206	William	Gietz
205	Shelley	Higgins
204	Hermann	Baer
203	Susan	Mavris
202	Pat	Fay

206 부터 내림차순으로 정렬됨

DISTINCT 중복 값 제거하고 고유값만 출력

```
1 SELECT job_id
2 FROM hr.employees;
```

AD_VP
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_MGR
HR_REP
IT_PROG
IT_PROG
IT_PROG

BEFORE 중복값 있음

```
1 SELECT DISTINCT job_id
2 FROM hr.employees;
```

AD_VP
FI_ACCOUNT
FI_MGR
HR_REP
IT_PROG
MK_MAN
MK_REP
PR_REP

AFTER 고유값만 출력

▼ ALIAS / AS - 별명(별칭) 사용하기

주의사항

- 별명은 열이름을 임시로 보여줄 때만 바꾸어서 표현한다.
- 즉 원래 열이름 자체는 바꾸지 않는다.
- 이름을 바꿀 때 AS를 사용하며 생략도 가능하나 반드시 넣어야 할 때도 있다.
- 특히, 빈칸/특수기호/대소문자 등을 넣을 때는 반드시 ""를 사용해야만 한다.

```

1 SELECT first_name AS 성, last_name AS 이름
2 FROM hr.employees;

```

성	이름
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida

```

1 SELECT first_name AS 성, last_name AS "이름"
2 FROM hr.employees;

```

성	이름
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida

|| (버티컬 바 / 연결연산)

```

1 SELECT first_name|| ' ' ||last_name,
2        email|| '@naver.com'
3 FROM hr.employees;

```

FIRST_NAME ' ' LAST_NAME	EMAIL '@NAVER.COM'
Ellen Abel	EABEL@naver.com
Sundar Ande	SANDE@naver.com
Mozhe Atkinson	MATKINSO@naver.com
David Austin	DAUSTIN@naver.com
Hermann Baer	HBAER@naver.com
Shelli Baida	SBAIDA@naver.com

+ - * / 열값 계산하여 출력 산술연산

```
1 SELECT employee_id 사원번호, first_name||' '||last_name 이름,  
2 salary 급여,  
3 salary + 500 추가급여,  
4 salary - 100 인하급여,  
5 salary * 1.1 보너스  
6 FROM hr.employees;
```

사원번호	이름	급여	추가급여	인하급여	보너스
100	Steven King	24000	24500	23900	26400
101	Neena Kochhar	17000	17500	16900	18700
102	Lex De Haan	17000	17500	16900	18700
103	Alexander Hunold	9000	9500	8900	9900

AS생략 가능하다, 마지막 , 없어야 함, ;세미클론 마지막 중요

▼ WHERE - 조건절

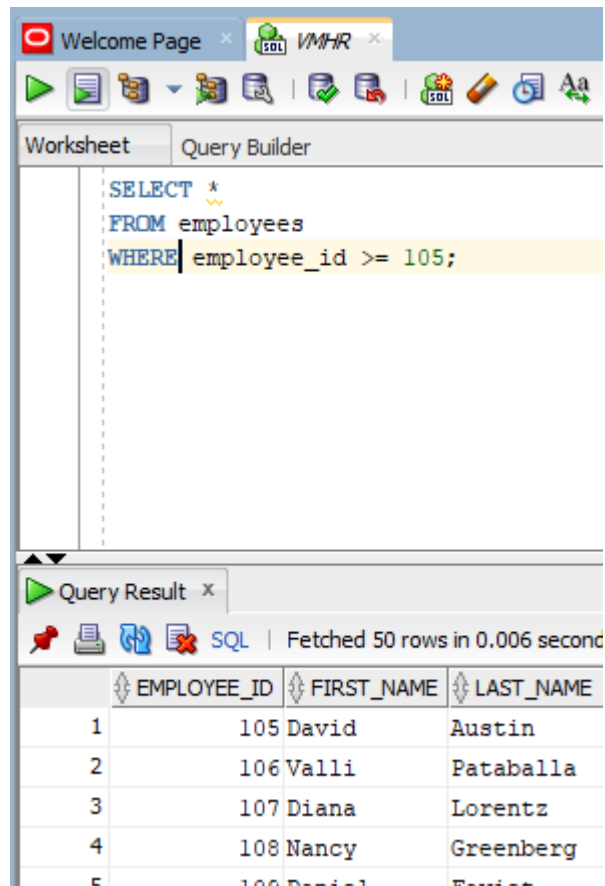
- where 절은 반드시 from 다음에 사용
- 원하는 조건만 골라서 조

기본형

SELECT 열이름 1, 열이름 2

FROM 테이블 이름

WHERE 조건;



WHERE절 뒤에 원하는 조건을 작성하면 그에 걸맞는 결과값이 출력된다.

- 복잡한 조건 적용하기

BETWEEN a AND b

IN(1,2,3,4)

LIKE

'%도' → '도'로 끝나는 값 다 찾기 (ex경기도/제주도 ...)

%@% → 포함하기만 하면 다 찾기

'서울%' → '서울'로 시작하는 말 다 찾기

is NULL null 찾기

BETWEEN a AND b

Worksheet Query Builder

```
SELECT *
FROM employees
WHERE salary BETWEEN 10000 AND 20000;
```

Query Result x

SQL | All Rows Fetched: 18 in 0.005 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000
2	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000
3	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-02	FI_MGR	12008
4	114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-02	PU_MAN	11000
5	145	John	Russell	JRUSSEL	011.44.1344.429268	01-OCT-04	SA_MAN	14000
6	146	Karen	Partners	KPARTNER	011.44.1344.467268	05-JAN-05	SA_MAN	13500

10000과 20000사이 전부





IN(1,2,3,4)

Worksheet

Query Builder

```
SELECT *  
FROM employees  
WHERE salary IN(10000, 13000, 17000);
```

Query Result x

    SQL | All Rows Fetched: 7 in 0.003 seconds

NAME	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT
1 CHAIR	515.123.4568	21-SEP-05	AD_VP	17000	
2 HAAS	515.123.4569	13-JAN-01	AD_VP	17000	
3 CLERK	011.44.1344.129268	30-JAN-05	SA_REP	10000	
4 KING	011.44.1345.429268	30-JAN-04	SA_REP	10000	
5 DECK	011.44.1343.829268	23-MAR-06	SA_REP	10000	
6 RISTE	515.123.5555	17-FEB-04	MK_MAN	13000	
7 ER	515.123.8888	07-JUN-02	PR_REP	10000	


10000, 13000, 17000 포함하는





LIKE

Worksheet

Query Builder

```
SELECT *  
FROM employees  
WHERE job_id LIKE 'AD%';
```










 Query Result x





    SQL | All Rows Fetched: 4 in 0.003 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
1	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD_ASST
2	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES
3	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP
4	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP

AD로 시작하는 것 모두 찾음

Welcome PageVMHR





WorksheetQuery Builder

```
SELECT *  
FROM employees  
WHERE job_id LIKE 'AD_';
```

Query Result

SQL | All Rows Fetched: 2 in 0.007 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD_ASST	4400
2	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000

_는 글자수를 의미한다

IS NULL 빈칸찾기

Worksheet Query Builder

```
SELECT *
FROM employees
WHERE manager_id IS NULL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0.006 seconds

ME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID
1	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	(null)	(null)

매니저 값이 없는 것 = 사장 / 빈칸 찾기