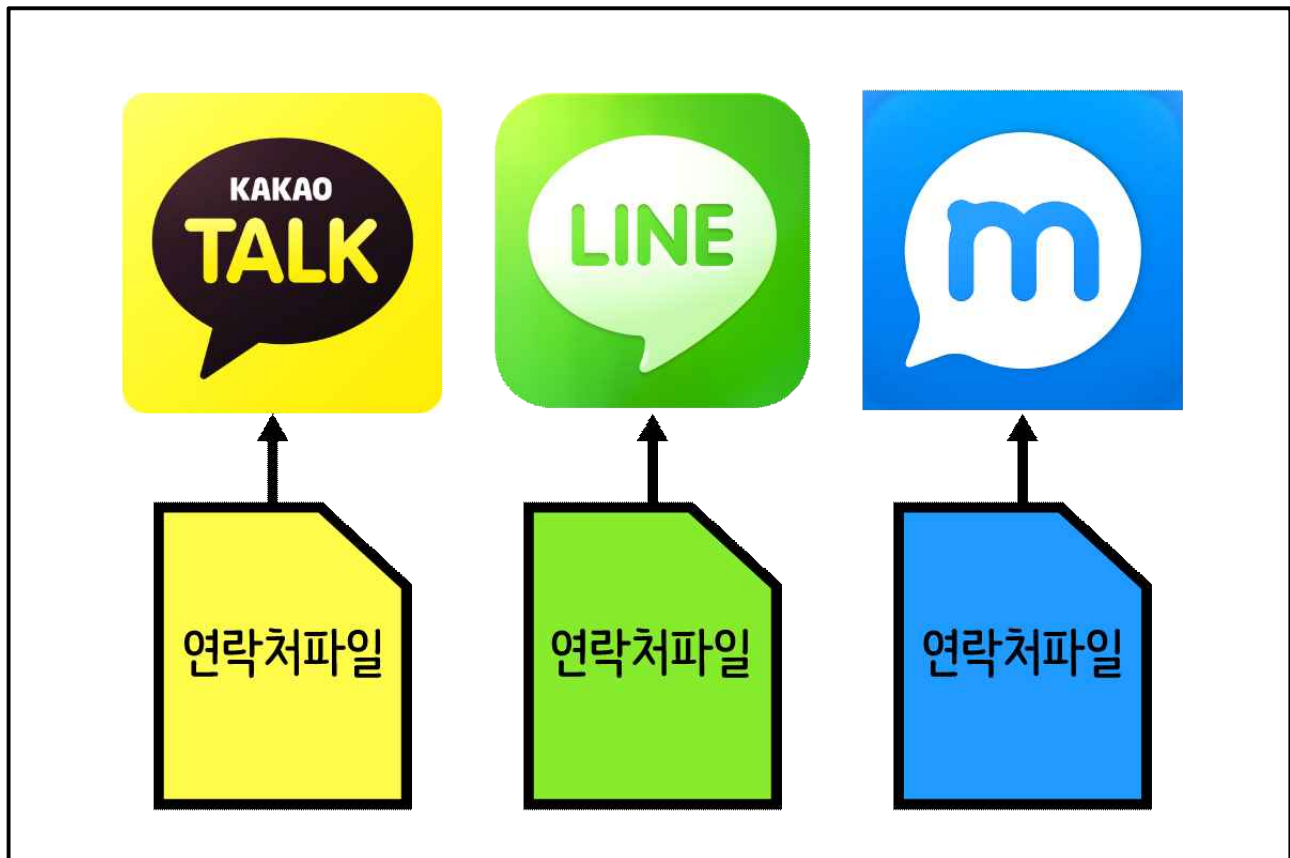


1. 파일시스템의 문제점

- 데이터 종속 : 응용프로그램과 데이터간 종속관계가 있음



- 데이터 중복시의 문제점

일관성	여러 개의 데이터가 모두 하나의 사실을 나타낸다면 논리적으로 그 내용이 모두 동일하나, 데이터 중복이 있으면 그 동일성을 유지하기가 어렵다.
보안성	동일 수준의 보안이 유지되어야 함에도 같은 데이터가 중복되어 있다면 모두 똑같은 수준의 보안을 유지하기 어렵다.
경제성	데이터를 중복해서 저장하기 위해서는 추가적으로 저장 공간에 대한 비용이 더 들게 된다.
무결성	데이터가 만족해야 할 제약조건이 정확성을 유지해야 한다.

2. 데이터베이스란?

- 데이터베이스는 유용한 데이터의 집합
- 유용한 데이터란 사용자가 원하는 정보를 쉽게 찾을 수 있어야 함
- 정보의 검색뿐 아니라, 수정, 삭제도 용이함

3. 데이터베이스의 정의

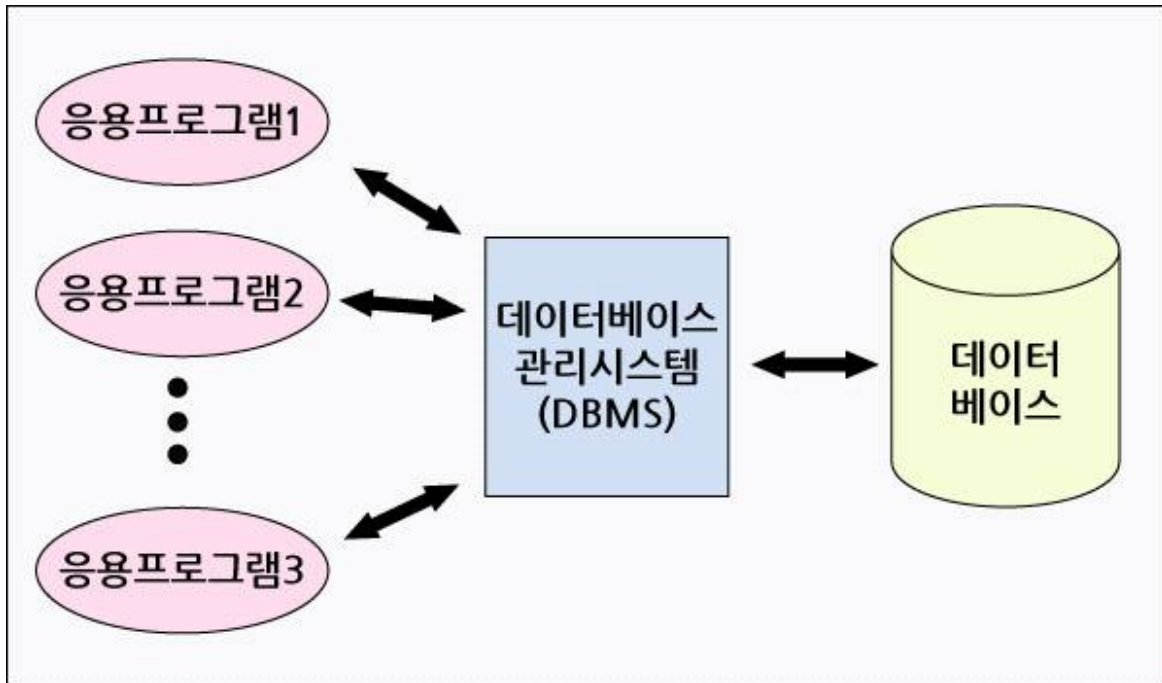
- 통합된 데이터 : 똑같은 데이터가 원칙적으로 중복되어 있지 않다는 것을 말함
- 저장된 데이터 : 컴퓨터가 접근할 수 있는 기억장치에 저장된 데이터
- 운영 데이터 : 존재 목적이 명확하고 유용성을 지니고 있는 데이터를 말함
- 공용 데이터 : 여러 사용자들이 서로 다른 목적으로 사용하는 공유 가능한 데이터를 말함

4. 데이터베이스의 특징

- 실시간 접근성
- 지속적인 변화
- 동시 공유
- 내용에 대한 참조

5. DBMS란(DataBase Management System)?

- 데이터베이스를 편리하게 저장하고 효율적으로 관리하고 검색할 수 있는 환경을 제공하는 시스템 소프트웨어를 DBMS라고 함



6. 데이터베이스의 종류

1) 관계형 데이터베이스(Relational Database)

- 2차원 구조의 모델을 기반으로 함
- 데이터의 무결성, 트랜잭션 처리 등 데이터베이스 관리 시스템으로써의 뛰어난 성능
- 질의어(Query Language)를 사용한 데이터 접근 방법
- 가장 많이 사용함

	성명(칼럼)	주소(칼럼)	전화번호(칼럼)	
	성 명	주 소	전화번호	
고객정보(로우)	김태은	서 울	010-123-4567	고객 테이블
고객정보(로우)	이은정	부 산	010-333-4567	
고객정보(로우)	조진이	대 구	010-555-4567	

2) 객체형 데이터베이스

3) 객체관계형 데이터베이스

4) 데이터베이스 종류에 따른 차이점

구 분	관계형 데이터베이스	객체지향 데이터베이스	객체관계형 데이터베이스
데이터 모델	문자,숫자,날짜등 단순정보만 지원	사용자 정의 타입, 비정형정보 타입 지원	사용자 정의 비정형정보 타입 지원
대규모 정보처리	우수	보통	우수
안정성	우수	보통	우수
장 점	사용하기 쉽고, 편리 하고 안정적	복잡한 구조의 정보모델링 가능	관계형의 안정성에 객 체지향 모델링 장점을 추가
단 점	확장성 부족, 복합적 인정보표현 어려움	안정성과 성능이 떨어 짐	

7. SQL이란?

■ Structured Query Language의 약자

■ RDBMS에서 데이터베이스에 질의, 수정, 삭제 등의 작업을 하는 언어의 표준

■ SQL의 종류

1) 데이터 정의어(DDL)

데이터베이스 관리자나 응용 프로그래머가 데이터베이스의 논리적 구조를 정의하기 위한 언어로 데이터 디렉터리에서 저장됨

2) 데이터 조작어(DML)

데이터베이스에 저장된 데이터를 조작하기 위해 사용되는 언어로, 검색, 추가, 삭제, 갱신 작업을 수행함

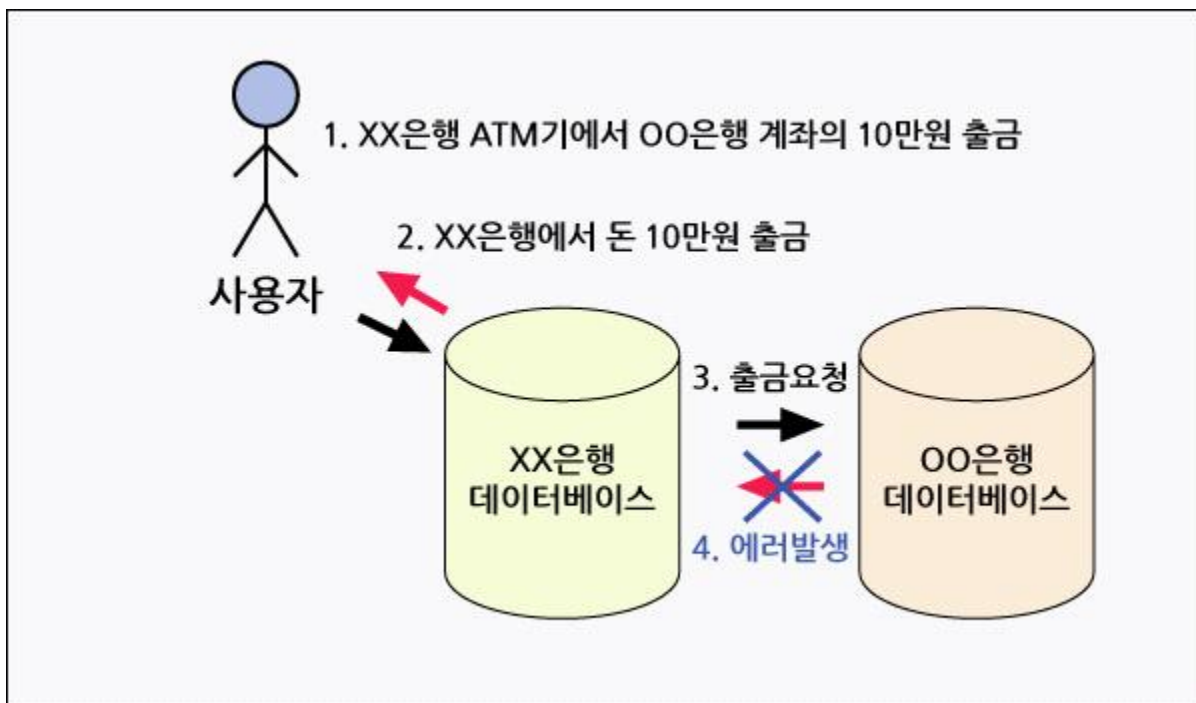
3) 데이터 제어어(DCL)

데이터에 대한 접근 권한 부여 등의 데이터베이스 시스템의 트랜잭션을 관리하기 위한 목적으로 사용되는 언어

유 형	명 령 문
DQL(질의어)	SELECT(데이터 검색시 사용)
DML(데이터 조작어)	INSERT(데이터 입력) UPDATE(데이터 수정) DELETE(데이터 삭제)
DDL(데이터 정의어)	CREATE(데이터베이스 객체 생성) ALTER(데이터베이스 객체 변경) DROP(데이터베이스 객체 삭제) RENAME(데이터베이스 객체 이름 변경) TRUNCATE(데이터베이스 객체 저장공간 삭제)
TCL(트랜잭션 처리어)	COMMIT(트랜잭션의 정상적인 종료 처리) ROLLBACK(트랜잭션 취소) SAVEPOINT(트랜잭션 내의 임시 저장점 설정)
DCL(데이터 제어어)	GRANT(데이터베이스에 대한 일련 권한 부여) REVOKE(데이터베이스에 대한 일련 권한 취소)

8. 트랜잭션이란?

- 트랜잭션이란 단어의 본래 뜻은 '거래'라는 의미
- 예를 들어 돈을 주었는데 물건을 받지 못한다면, 그 거래는 이루어 지지 못하고 원상태로 복구되어야 함
- 이와 같이 쪼갤 수 없는 하나의 처리 행위를 원자적 행위라고 함
- 여기서 쪼갤 수 없다는 말의 의미는 실제로 쪼갤 수 없다기 보다는 만일 쪼개질 경우 시스템에 심각한 오류를 초래할 수 있다는 것



9. 오라클이란?

- 1977년 래리 앨리슨이 설립
- CIA 프로젝트 이름이 '오라클'이었고 이 이름을 차용하여 오라클이라 명명함
- 8i, 9i의 I는? : internet을 의미
- 10g, 11g의 G는? : grid를 의미함
- 12c? : cloud를 의미함

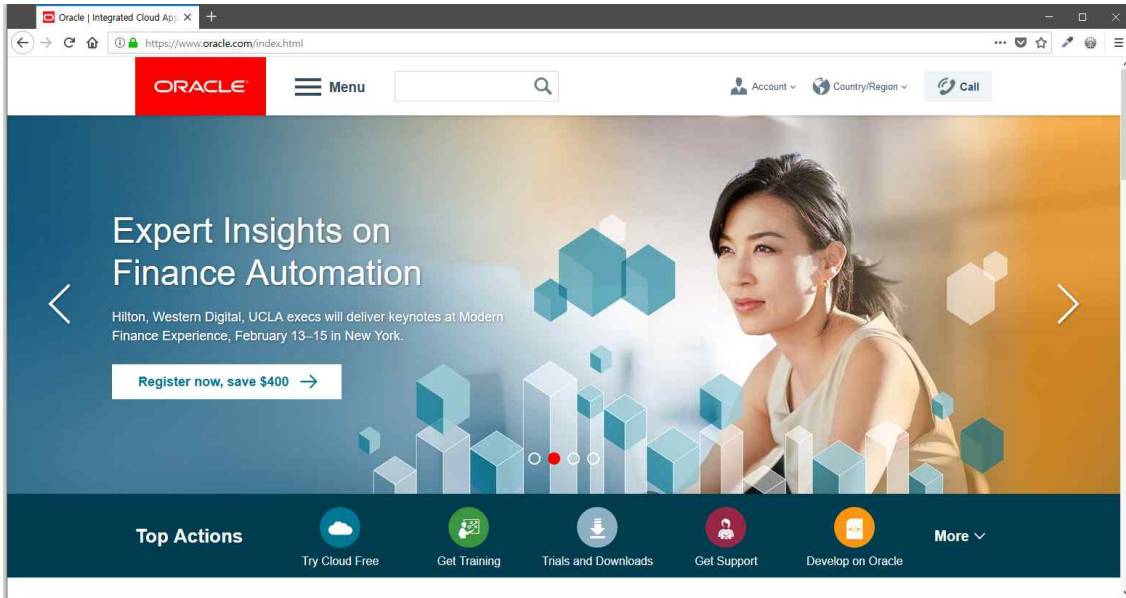
- 1) 그리드컴퓨팅 : 위치상으로는 분리되어 있지만 원거리 통신망으로 연결되어 있는 여러 대의 컴퓨터의 유휴자원을 가상화하여 마치 하나의 대용량 고성능 컴퓨터 인 것처럼 만들어 연산을 수행하는 기술을 의미함
- 2) 클라우드컴퓨팅 : 개인이 가진 단말기를 통해서는 주로 입/출력 작업만 이루어지고, 정보분석 및 처리, 저장, 관리, 유통 등의 작업은 클라우드라고 불리는 제3의 공간에서 이루어지는 컴퓨팅 시스템 형태

10. 오라클 에디션별 특징

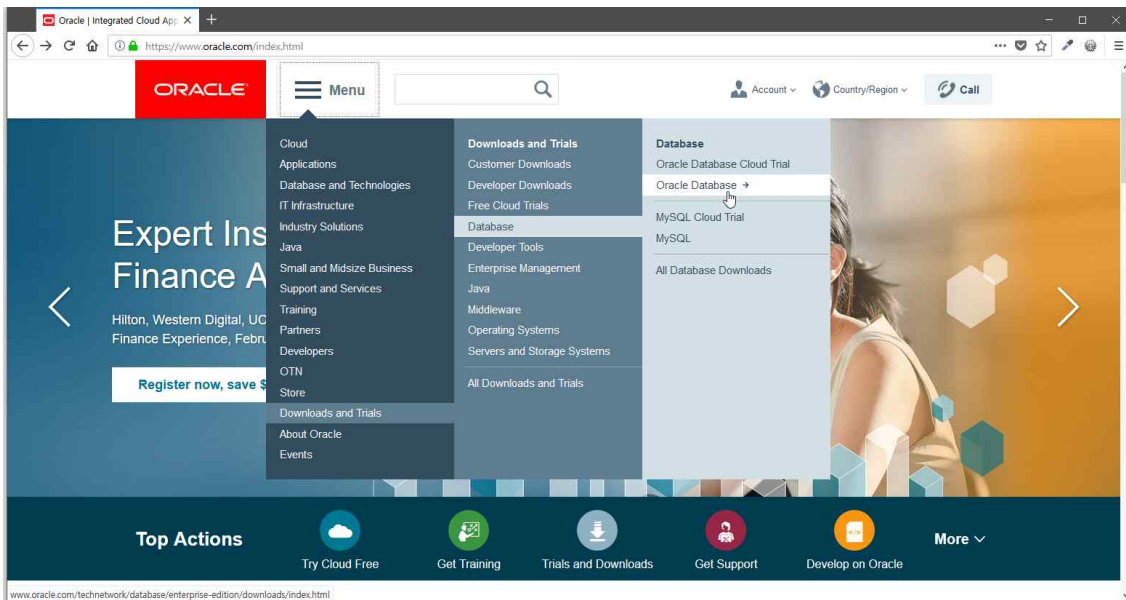
- 엔터프라이즈 에디션 : 오라클 데이터베이스의 기능을 모두 가진 에디션으로, 서버 CPU의 개수가 4개 이상일 경우 설치해야 함. 상용으로 사용되는 경우 가장 많이 사용
- 스탠다드 에디션 : 엔터프라이즈보다는 기능 제한이 있음. CPU 2개 이하일 경우 가능
- 익스프레스 에디션 : 무료이므로 호스트당 1개의 CPU, 1GB의 메모리, 최대 저장 데이터 4GB 제한됨
- 퍼스널 에디션 : 개인용 에디션으로 한 명의 사용자만 등록하여 사용 가능함
- 오라클 데이터베이스 라이트 : 모바일용으로 사용하는 데이터베이스

11. Oracle 11g Express Edition 의 설치

- oracle 홈페이지



- Download and Trials -> Database -> Oracle Database



- Oracle Database 11g Express Edition



- Accpet

Oracle Database Express Edition 11g Release 2

June 4, 2014

You must accept the [OTN License Agreement for Oracle Database Express Edition 11g Release 2](#) to download this software.

☒ Accept License Agreement | ☐ Decline License Agreement

- ↓ [Oracle Database Express Edition 11g Release 2 for Windows x64](#)
- Unzip the download and run the DISK1/setup.exe
- ↓ [Oracle Database Express Edition 11g Release 2 for Windows x32](#)
- Unzip the download and run the DISK1/setup.exe
- ↓ [Oracle Database Express Edition 11g Release 2 for Linux x64](#)
-Unzip the download and the RPM file can be installed as normal

- x64 다운로드

Oracle Database Express Edition 11g Release 2



June 4, 2014



Thank you for accepting the License Agreement, you may now download this software.

- ↓ [Oracle Database Express Edition 11g Release 2 for Windows x64](#)
- Unzip the download and run the DISK1/setup.exe
- ↓ [Oracle Database Express Edition 11g Release 2 for Windows x32](#)
- Unzip the download and run the DISK1/setup.exe
- ↓ [Oracle Database Express Edition 11g Release 2 for Linux x64](#)
-Unzip the download and the RPM file can be installed as normal


- 회원가입이 필요함

로그인

 사용자 이름 

 암호 

로그인

계정 만들기 

© Oracle | 이용 약관 | 개인정보 보호정책

- 오라클 계정 만들기

Oracle 계정 만들기

Oracle 계정이 이미 있으십니까? [로그인](#)

이메일 주소*

비밀번호*

비밀번호 재입력*

국가* 대한민국 ▼

이름*

이름(영문)*

직책*

직장 전화*

회사명*

주소*

구/군/시*

시/도* -선택- ▼

우편 번호*

이메일 주소가 사용자 이름입니다.

비밀번호는 이메일의 일부와 일치하지 않는 최소 하나 이상의 숫 두 가져야 하며 8자 이상이어야 합니다.

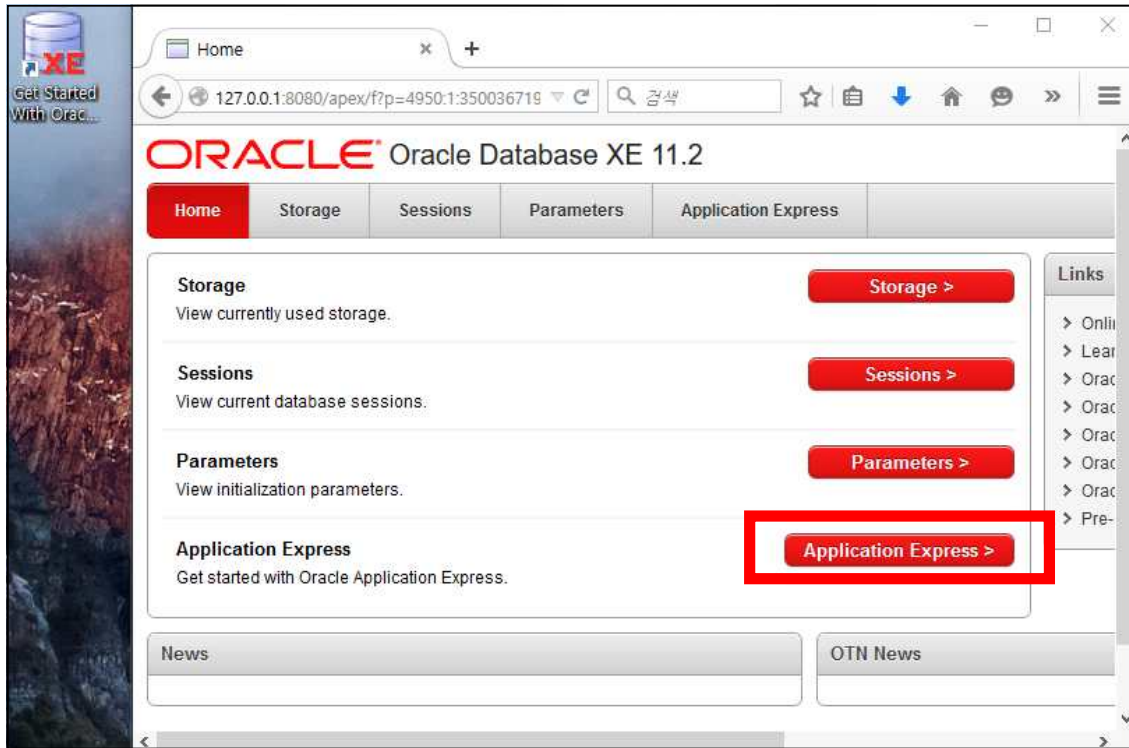
☐ 예, Oracle 제품, 서비스 및 이벤트에 대한 이메일을 보내 주십시오.

[모든 마케팅 소식 수신 거부](#) [구독 취소](#)

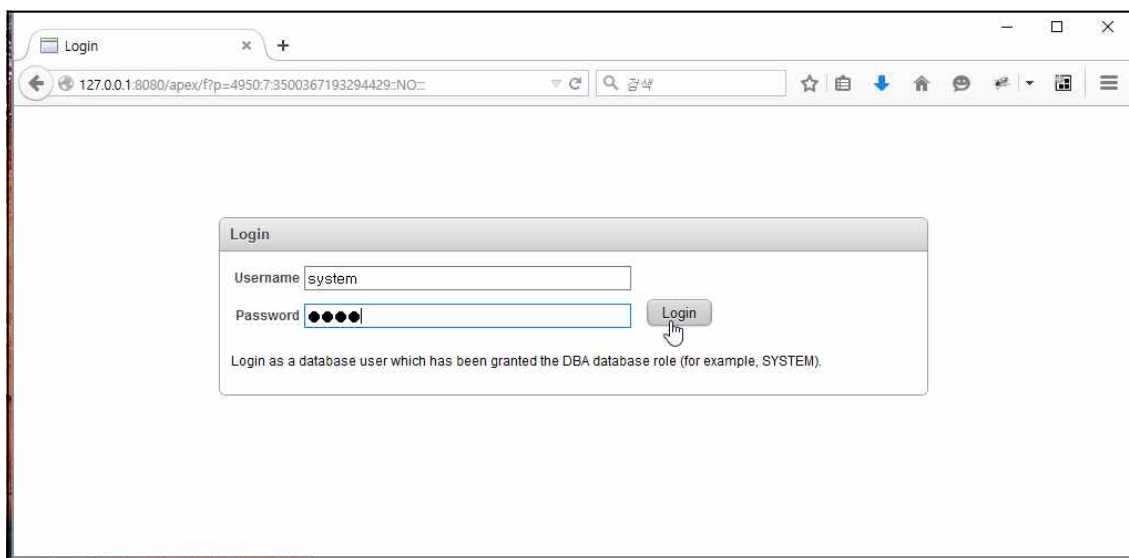
- 로그인후 다운로드

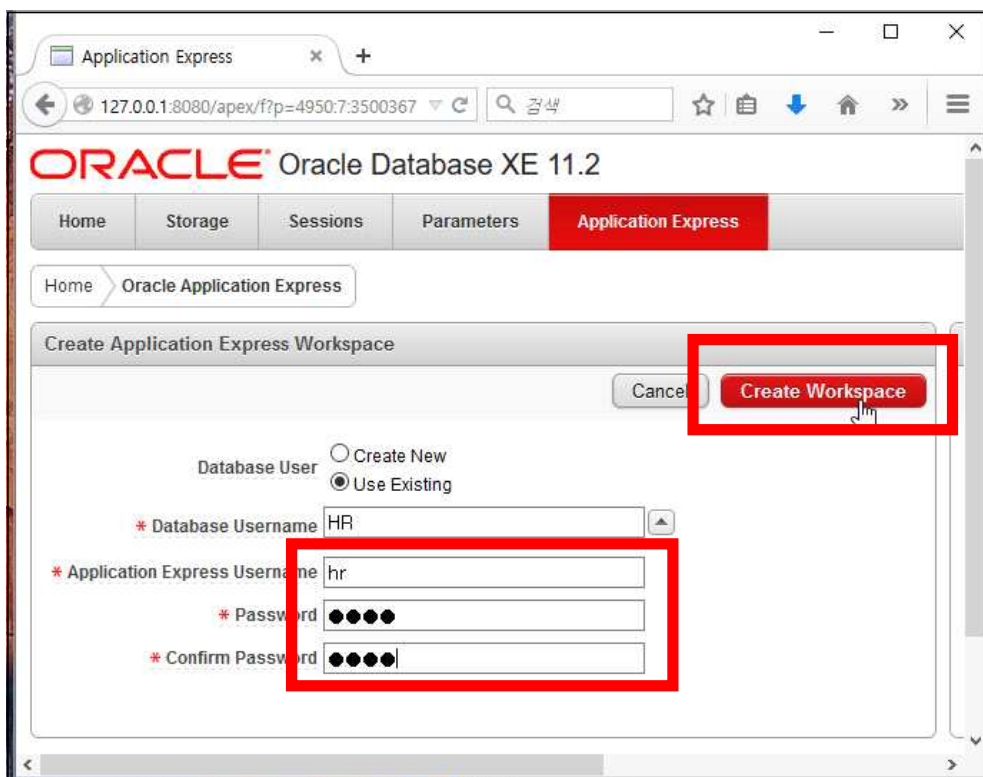
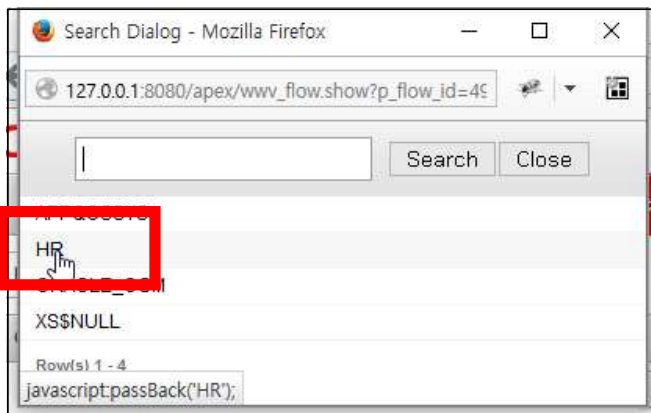
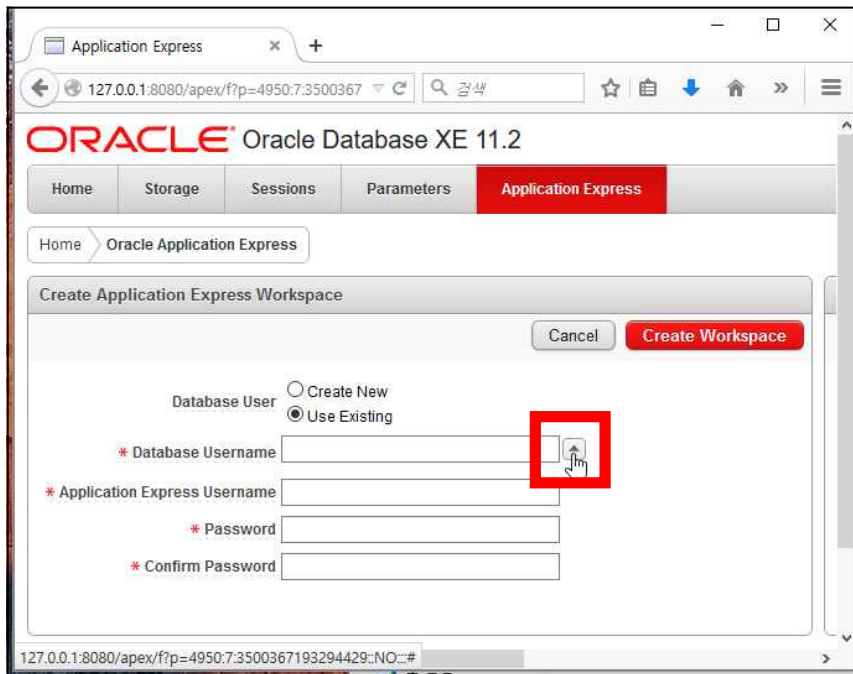
■ 데이터베이스 홈페이지 시작

- Application Express 선택

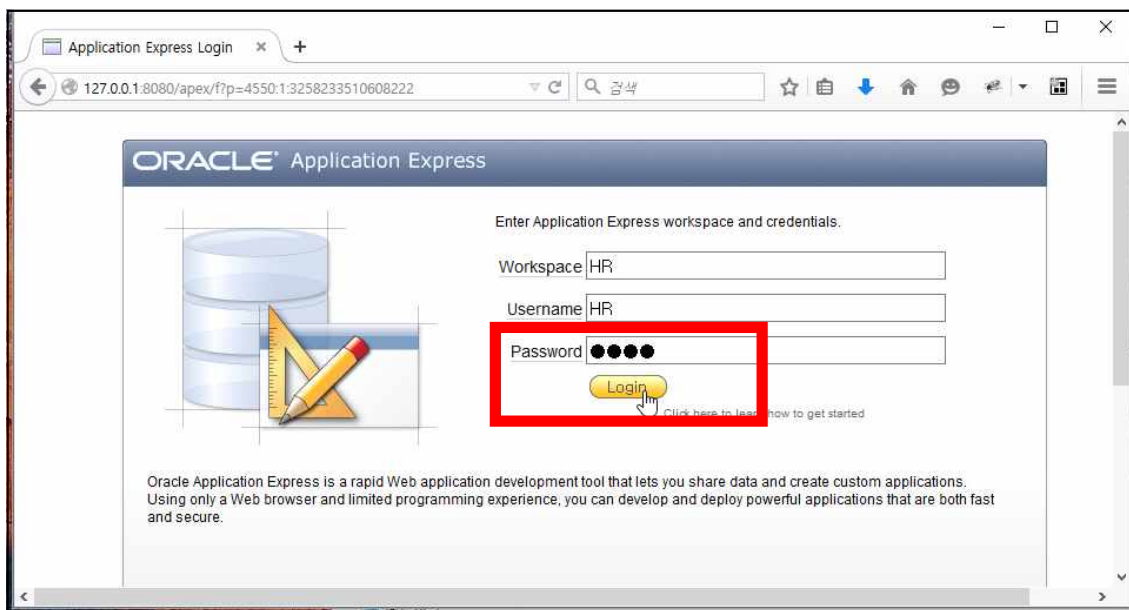
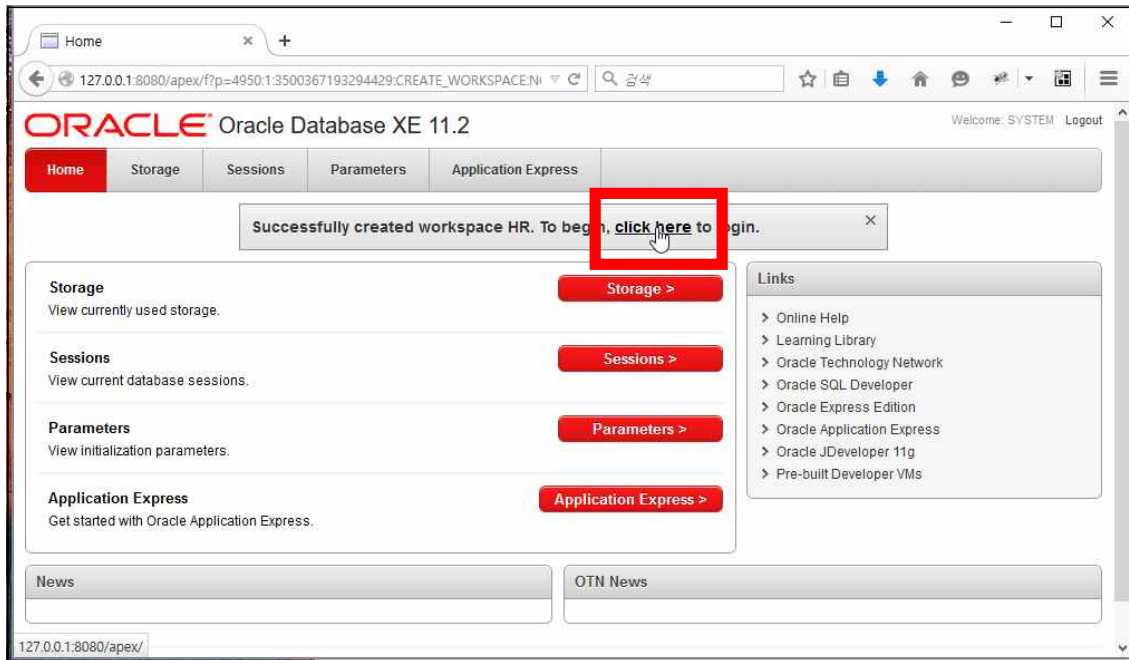


■ 이름 : system, 암호 (1111) 입력 후 로그인 후 hr계정 선택





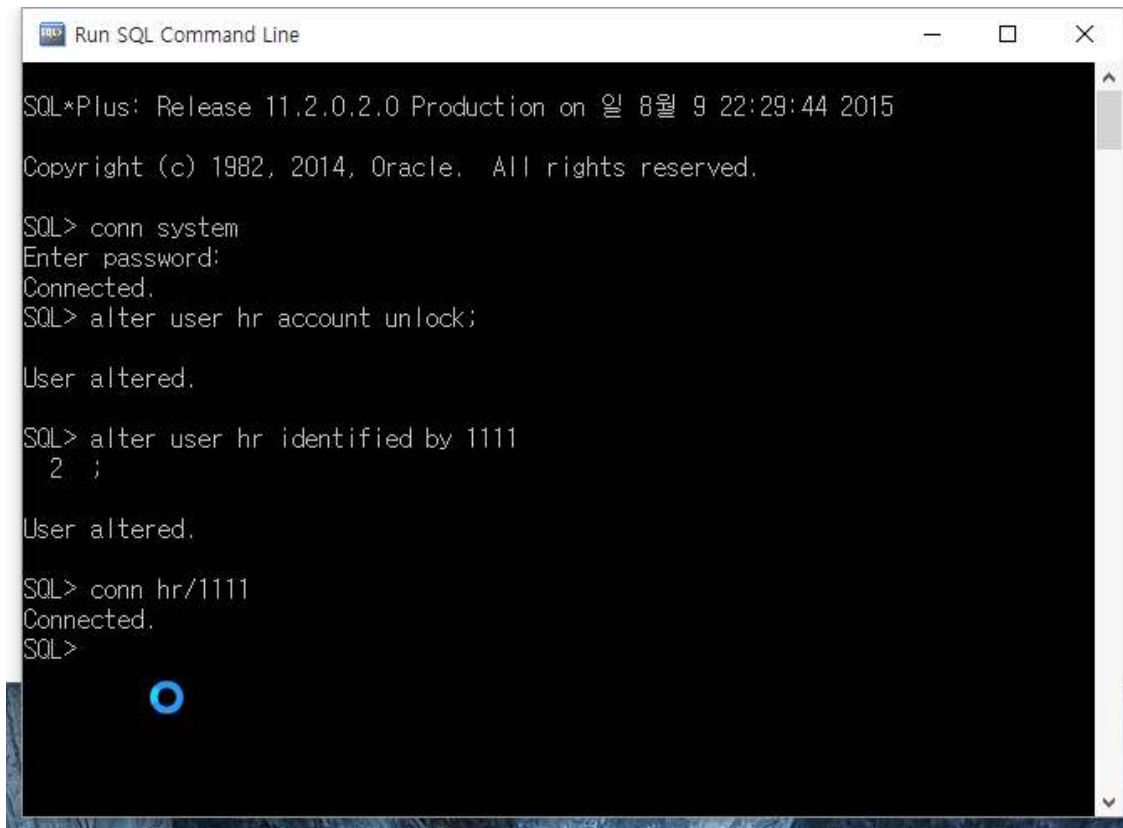
■ hr 계정으로 다시 로그인 하기



12. CMD 창으로 연결하는 법

- 윈도우 시작 버튼 -> 모든 프로그램 -> Oracle Database 10g Express Edition
-> SQL 명령줄 실행
- 'conn system/(암호) as sysdba' 로 연결
- alter user hr account unlock; 으로 lock 해제

- alter user hr identified by (암호명) 으로 암호 입력
- conn hr/(암호) 로 연결
- select * from departments; 로 테이블 데이터 보기



```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on 일 8월 9 22:29:44 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> conn system
Enter password:
Connected.
SQL> alter user hr account unlock;

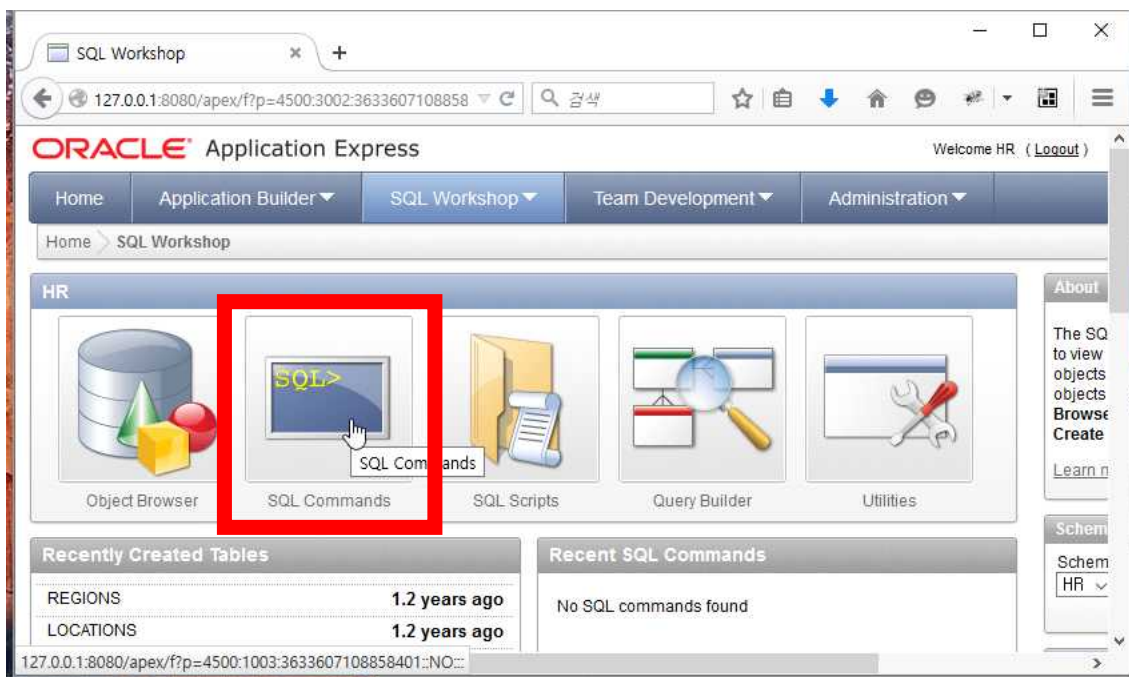
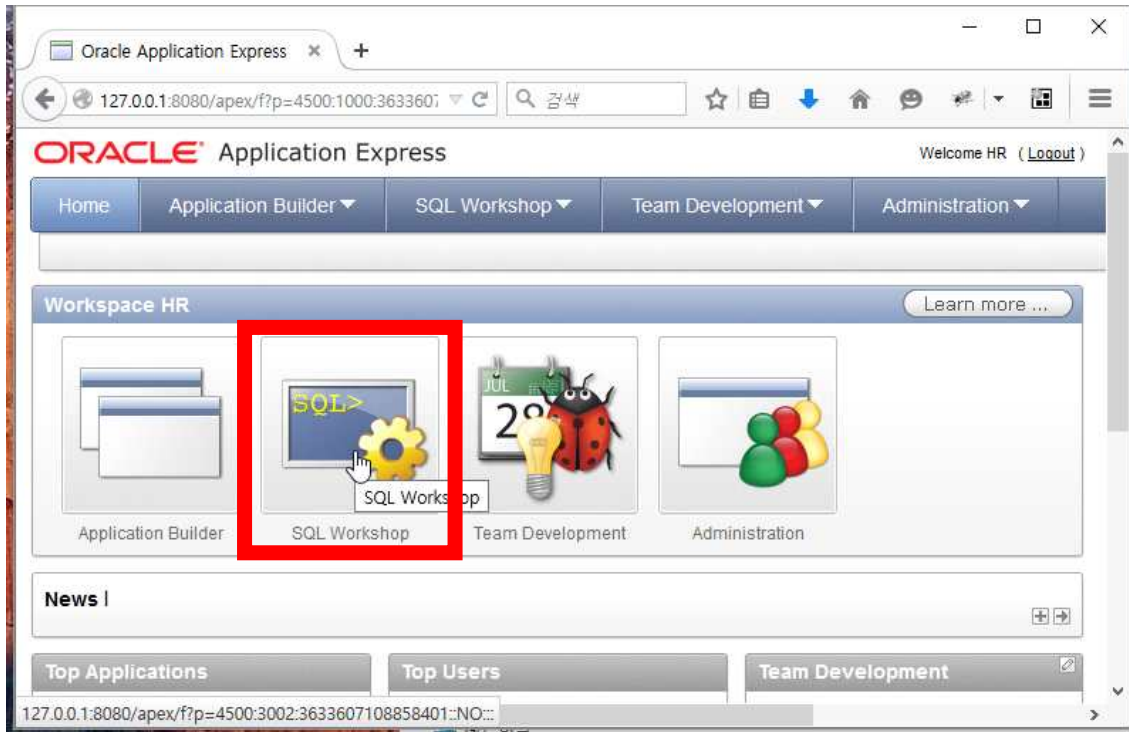
User altered.

SQL> alter user hr identified by 1111
2 ;

User altered.

SQL> conn hr/1111
Connected.
SQL>
```


13. SQL의 기본



■ 데이터의 구조를 살펴보기 위한 DESC : desc 테이블명;

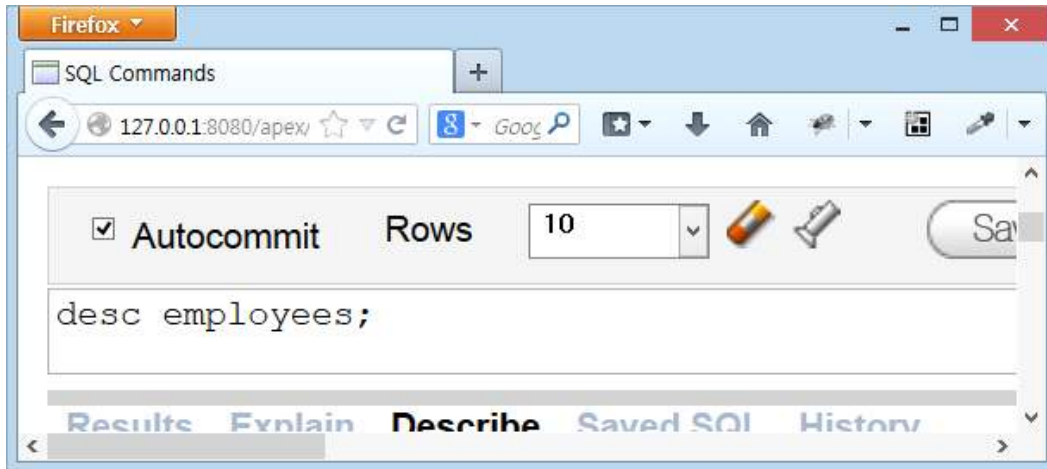


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEES	EMPLOYEE_ID	NUMBER	-	6	0	1	-	-	Primary key of employees table.
	FIRST_NAME	VARCHAR2	20	-	-	-	✓	-	First name of the employee. A not null column.
	LAST_NAME	VARCHAR2	25	-	-	-	-	-	Last name of the employee. A not null column.
	EMAIL	VARCHAR2	25	-	-	-	-	-	Email id of the employee
	PHONE_NUMBER	VARCHAR2	20	-	-	-	✓	-	Phone number of the employee; includes country code and area code
	HIRE_DATE	DATE	7	-	-	-	-	-	Date when the employee started on this job. A not null column.
	JOB_ID	VARCHAR2	10	-	-	-	-	-	Current job of the employee; foreign key to job_id column of the jobs table. A not null column.
	SALARY	NUMBER	-	8	2	-	✓	-	Monthly salary of the employee. Must be greater than zero (enforced by constraint emp_salary_min)
	COMMISSION_PCT	NUMBER	-	2	2	-	✓	-	Commission percentage of the employee; Only employees in sales department eligible for commission percentage
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	Manager id of the employee; has same domain as manager_id in departments table. Foreign key to employee_id column of employees table. (useful for reflexive joins and CONNECT BY query)
	DEPARTMENT_ID	NUMBER	-	4	0	-	✓	-	Department id where employee works; foreign key to department_id column of the departments table

■ 오라클의 데이터형

1) number형 : NUMBER(precision, scale) 둘 다 생략가능

- precision는 소수점을 포함한 전체 자리수
- scale은 소수점 이하 자리수

2) date형 : 시간 데이터를 저장하기 위한 데이터 형

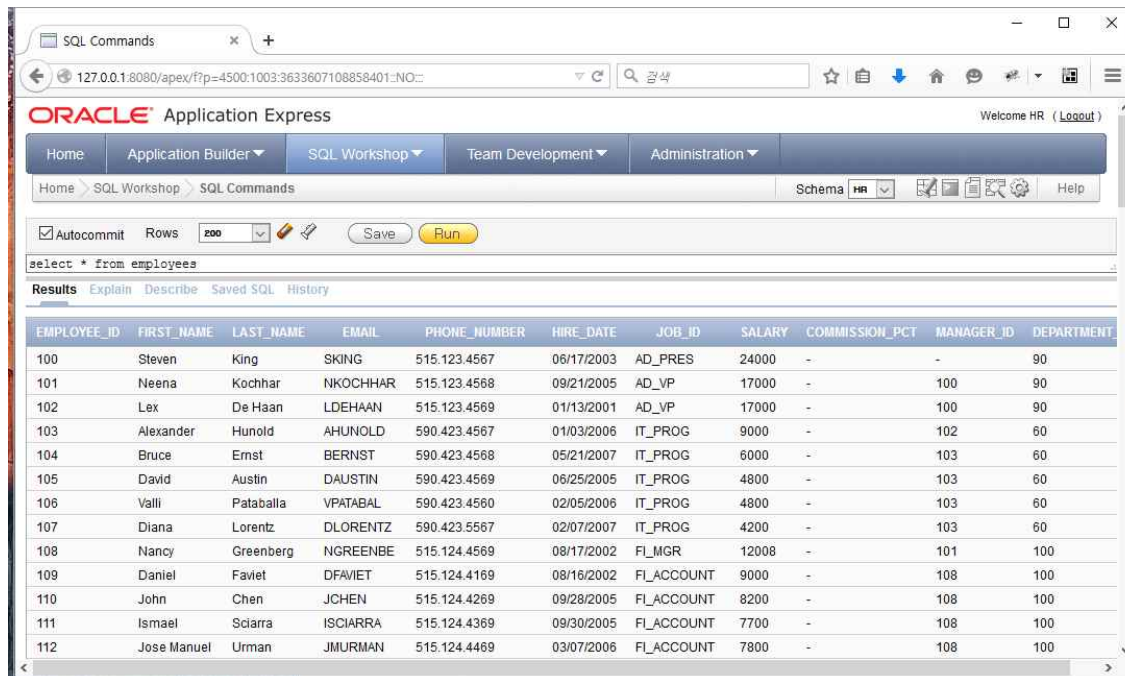
(한글판과 영어판 출력형식 다름)

3) varchar2형 : 가변적인 길이의 문자열을 저장하기 위해 제공

4) char형 : 고정적인 길이의 문자열을 저장하기 위해 제공

■ 기본적인 SELECT문

- 1) SELECT * FROM employees;
- 2) '*'는 모든 컬럼명을 조회함(특정 컬럼만 조회 가능)
- 3) FROM 다음에 보고자 하는 테이블의 이름을 기술함
- 4) 결과

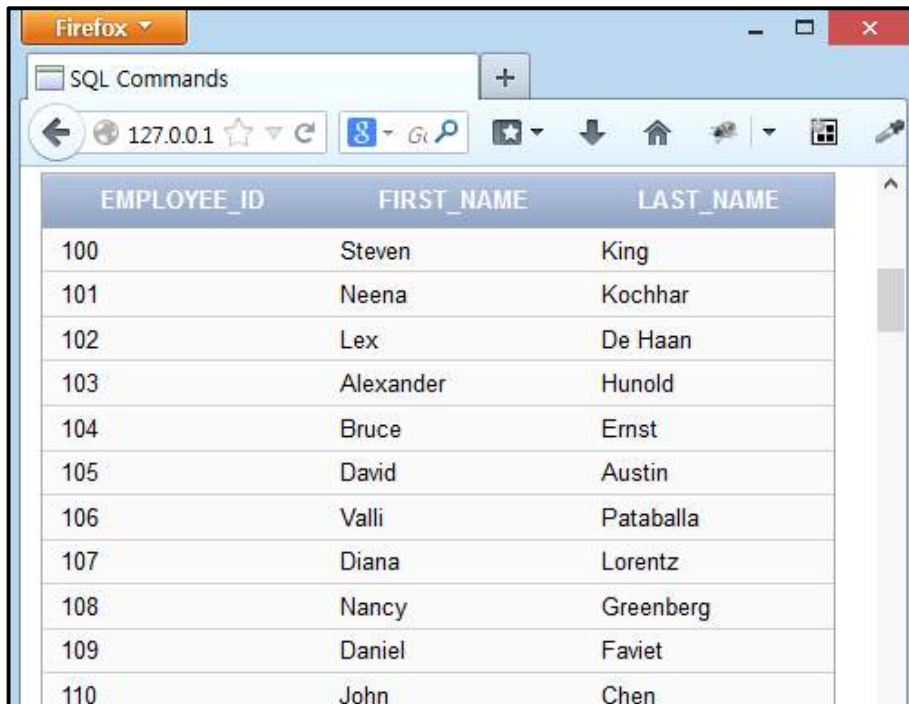


The screenshot shows the Oracle Application Express interface. The SQL Commands window contains the query `select * from employees`. The results are displayed in a table with 11 columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT, MANAGER_ID, and DEPARTMENT. The table contains 12 rows of data.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT
100	Steven	King	SKING	515.123.4567	06/17/2003	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	09/21/2005	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	01/13/2001	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	01/03/2006	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	05/21/2007	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	06/25/2005	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	02/05/2006	IT_PROG	4800	-	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	02/07/2007	IT_PROG	4200	-	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	08/17/2002	FL_MGR	12008	-	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	08/16/2002	FL_ACCOUNT	9000	-	108	100
110	John	Chen	JCHEN	515.124.4269	09/28/2005	FL_ACCOUNT	8200	-	108	100
111	Ismael	Sciarra	ISCIARRA	515.124.4369	09/30/2005	FL_ACCOUNT	7700	-	108	100
112	Jose Manuel	Urman	JMURMAN	515.124.4469	03/07/2006	FL_ACCOUNT	7800	-	108	100

■ 특정 데이터만 보기

1) `SELECT employee_id, first_name, last_name FROM employees;`



EMPLOYEE_ID	FIRST_NAME	LAST_NAME
100	Steven	King
101	Neena	Kochhar
102	Lex	De Haan
103	Alexander	Hunold
104	Bruce	Ernst
105	David	Austin
106	Valli	Pataballa
107	Diana	Lorentz
108	Nancy	Greenberg
109	Daniel	Faviet
110	John	Chen

2) 예제 : 사원id(EMPLOYEE_ID)과 급여(SALARY)와 입사일자(HIRE_DATE)를 출력하세요.

■ 산술 연산자

종 류	예
+	<code>SELECT employee_id, salary + commision_pct FROM employees;</code>
-	<code>SELECT employee_id, salary - 100 FROM employees;</code>
*	<code>SELECT employee_id, salary * 2 FROM employees;</code>
/	<code>SELECT employee_id, salary / 2 FROM employees;</code>

1) 예제 : 연봉을 계산해보세요.(salary * 12)

(출력필드는 사번, 이름, 성, 연봉)

■ NULL이란?

- 1) 아직 정의되지 않은 값
- 2) 0 또는 공백과는 다름(0은 하나의 숫자, 공백은 하나의 문자)
- 3) 테이블을 생성할 때 null 허용에 대한 정의를 내릴 수 있음
- 4) null값을 포함하는 연산의 경우 결과는 null값임

■ 별칭(alias)

- 1) 컬럼의 이름이 컬럼에 대한 컬럼헤딩(heading)으로 출력됨
- 2) 컬럼이름으로 알기 어려운 경우 별칭을 부여할 수 있음
- 3) 'as' 로 별칭 부여
- 4) `SELECT employee_id, salary * 12 as "연봉" FROM employees;`

EMPLOYEE_ID	연봉
100	288000
101	204000
102	204000
103	108000
104	72000
105	57600
106	57600
107	50400
108	144000

14. 연산자

■ WHERE 절

```
SELECT 컬럼명  
FROM 테이블명  
WHERE 조건
```

■ 조건절의 구성 : ① 컬럼 ② 연산자 ③ 비교대상값

■ 비교 연산자

연산자	설명
=	같다
>	크다
<	작다
>=	크거나 같다
<=	작거나 같다
<>	같지 않다

■ 예제

```
SELECT *  
FROM employees  
WHERE department_id;
```

① 문제 : employees 테이블에서 급여가 2000이상 되는 사람을 출력하세요.

(출력할 열 : 직원번호, 이름, 급여, 고용일)

■ 문자열 데이터 조회

```
SELECT employee_id, last_name, first_name  
FROM employees  
WHERE last_name = 'Cambrault';
```

② 문제 : employees 테이블에서 직업이 ST_CLERK인 사람들을 모두 출력하세요.

(출력할 열 : 직원번호, 이름, 직업, 급여)

■ 날짜 데이터 조회

```
SELECT employee_id, last_name, first_name, hire_date  
FROM employees  
WHERE hire_date >= '1993/01/01';
```

③ 문제 : employees 테이블에서 1997년 이후에 고용된 사람들을 모두 구하세요.

(출력할 열 : 사원번호, 이름, 성, 고용일)

■ LIKE 연산자

```
WHERE 컬럼명 LIKE(연산자) '비교값(패턴)'
```

%	0 ~ n 개의 문자열을 대체함
_	단 1글자의 문자열을 대체함

■ 예제 : '%'를 사용하기

```
SELECT employee_id, last_name, first_name
FROM employees
WHERE last_name LIKE 'L%';
```

④ 문제 : 성이 's'로 끝나는 사람들을 표현하세요.

(출력할 열 : 사원번호, 이름, 성)

■ 예제2 : '_'를 사용하기

```
SELECT employee_id, last_name, first_name
FROM employees
WHERE last_name LIKE '_a%';
```

■ 예제3 : 가운데 '%'를 포함한 데이터 검색

```
SELECT employee_id, last_name, first_name
FROM employees
WHERE last_name LIKE '%a%';
```

⑤ 문제 : 이름에 'ab'가 있는 사람들을 출력하세요.

(출력할 열 : 직원번호, 이름, 성)

■ IN연산자(OR을 이용함)

```
SELECT employee_id, last_name, first_name, salary
FROM employees
WHERE salary = 17000 OR salary = 8000 OR salary = 2700;
```

■ IN연산자를 사용함

```
SELECT employee_id, last_name, first_name, salary
FROM employees
WHERE salary IN(17000, 8000, 2700);
```

⑥ 문제 : 성이 'Lorentz'이거나 'Popp'인 사람을 출력하세요.

(출력할 열 : 직원번호, 이름, 성)

■ BETWEEN AND 연산자(AND를 이용)

```
SELECT employee_id, last_name, first_name, salary
FROM employees
WHERE salary >= 2000 AND salary <= 5000;
```

■ BETWEEN AND 를 이용

```
SELECT employee_id, last_name, first_name, salary
FROM employees
WHERE salary BETWEEN 2000 AND 5000;
```

⑦ 문제 : commission_pct가 0.2와 0.3사이에 있는 사람들을 출력하세요.

(출력할 열 : 직원번호, 이름, 성, 커미션퍼센트)

■ AND 연산자

```
SELECT employee_id, last_name, first_name, job_id, department_id
FROM employees
WHERE department_id = 100 AND job_id = 'FI_MGR';
```

※ BETWEEN AND 는 같은 칼럼명, AND는 여러 칼럼 모두

⑧ 문제 : 이름이 'Nancy'인 사람의 밑에서 일하는 사람들 중에 salary가 8000이 넘는 사람들을 출력하세요.

(출력할 열 : 직원번호, 이름, 성, 월급, 매니저 id)

■ OR 연산자

```
SELECT employee_id, salary, hire_date, job_id
FROM employees
WHERE hire_date >= '2005/01/01' OR job_id = 'IT_PROG';
```

⑨ 문제 : 고용일이 2002년부터 2004년 사이인 사원이거나, salary가 20000이상의 사람들을 출력하세요. (출력할 열 : 직원번호, 이름, 성, 월급, 고용일)

■ NOT 연산자

```
SELECT employee_id, salary, last_name, first_name
FROM employees
WHERE NOT salary <= 10000;
```

⑩ 문제 : commission_pct 가 0.1, 0.3인 사람들을 출력하세요.

(출력할 열 : 직원번호, 이름, 성, 커미션퍼센트)

■ NOT IN 연산자

```
SELECT employee_id, last_name, first_name, commission_pct
FROM employees
WHERE commission_pct NOT IN(0.1, 0.3);
```

⑪ 문제 : 부서id가 10, 20, 30이 아닌 부서에 근무하는 사람들을 모두 출력하세요.(출력할 열 : 사원번호, 이름, 성, 부서번호)

■ NOT LIKE

```
SELECT employee_id, last_name, first_name
FROM employees
WHERE last_name NOT LIKE 'L%';
```

⑫ 문제 : 이름이 'A로 시작하지 않고, 's'로 끝나지 않는 사람들을 모두 출력하세요.(출력할 열 : 사원번호, 이름)

■ NOT BETWEEN A AND B

```
SELECT employee_id, last_name, first_name, salary
FROM employees
WHERE salary NOT BETWEEN 2000 AND 5000;
```

⑬ 문제 : 연봉이 5만불에서 20만불 사이가 아닌 사람들을 출력하세요.
(출력할 열 : 사원번호, 이름, 연봉)

■ IS NULL, IS NOT NULL

```
SELECT employee_id, last_name, first_name, salary
FROM employees
WHERE manager_id IS NULL;
```

※ NULL은 '='이 아니고 'IS'를 사용함

```
SELECT employee_id, last_name, first_name, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL;
```


■ ORDER BY절

```
SELECT 컬럼명들  
FROM 테이블명  
WHERE 조건식  
ORDER BY 컬럼명 (정렬기준:DESC, ASC);
```

⑭ 문제 : 일찍 고용된 사람들부터 성을 A~Z순으로 출력하세요.

(출력할 열 : 직원번호, 고용일, 성)

■ job_id 출력

```
SELECT job_id  
FROM employees;
```

■ job_id 중복제거(DISTINCT)

```
SELECT DISTINCT job_id  
FROM employees;
```

⑮ 문제 : 부서id를 중복없이 표현하세요.

■ 이름과 성을 붙여서 표현

```
SELECT employee_id, last_name || ' ' || first_name as ename  
FROM employees
```

⑩ 문제 : 아래쪽 스크린샷처럼 결과를 만드세요. (아래쪽 생략)

사람마다 아이디 찾기
Zlotkey Eleni의 employee_id는 149 이다.
Whalen Jennifer의 employee_id는 200 이다.
Weiss Matthew의 employee_id는 120 이다.
Walsh Alana의 employee_id는 196 이다.
Vollman Shanta의 employee_id는 123 이다.
Vishney Clara의 employee_id는 162 이다.
Vargas Peter의 employee_id는 144 이다.
Urman Jose Manuel의 employee_id는 112 이다.
Tuvault Oliver의 employee_id는 155 이다.
Tucker Peter의 employee_id는 150 이다.
Tobias Sigal의 employee_id는 117 이다.

■ 복습

- 1) 캐나다의 contry id는?
- 2) 60번 부서의 이름은?
- 3) 부사장은 몇 명이고 이름은?
- 4) SA_MAN의 직업 이름은?
- 5) 1500번 로케이션의 도시이름은?
- 6) 지역 3번은 어느 지역인가요?

2. SQL 함수의 특징

- 자원들의 계산을 수행할 수 있다.(최대값,최소값....)
- 개별적인 자원을 수정할 수 있다.
- 행의 그룹에 대해 결과를 조작할 수 있다.
- 출력 형식을 지정할 수 있다.

3. 단일 행 함수 : 반환되는 결과 행 각각의 대해 수행하는 함수

문자형 함수	문자를 받고 문자와 숫자 값 모두를 반환할 수 있음
숫자형 함수	숫자를 받고 숫자를 반환함
날짜형 함수	날짜형에 대해 수행하고 숫자를 반환하는 MONTH_BETWEEN 함수를 제외하고 모두 날짜 데이터 반환
일반적인 함수	nvl, decode

- DUAL 테이블 : select문장은 FROM이 꼭 필요로 함

```
SELECT 10 * 20  
FROM departments;
```

```
SELECT 10 * 20  
FROM dual;
```

```
DESC SYS.DUAL;
```

```
SELECT sysdate  
FROM dual;
```

4. 문자형 함수

■ 변환 함수

1) LOWER : 값을 모두 소문자로 변환

```
SELECT LOWER('WELCOME TO ORACLE')  
FROM dual;
```

2) UPPER : 값을 모두 대문자로 변환

① 문제 : 'Welcome To oracle'을 대문자로 변환하세요.

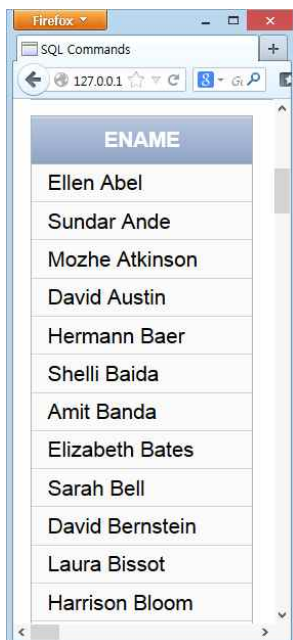
3) INITCAP : 값의 첫문자만 대문자로 변환

② 문제 : 'WELCOME TO ORACLE'을 첫문자만 대문자로 변환시키세요.

■ 문자 조작 함수

1) CONCAT : 두 문자열을 연결

```
SELECT CONCAT('WELCOME TO ', 'ORACLE')  
FROM dual;
```



ENAME
Ellen Abel
Sundar Ande
Mozhe Atkinson
David Austin
Hermann Baer
Shelli Baida
Amit Banda
Elizabeth Bates
Sarah Bell
David Bernstein
Laura Bissot
Harrison Bloom

③ 문제 : CONCAT함수만을 사용하여 왼쪽과 같이 나오게 함

2) SUBSTR : 문자열에서 문자열 일부만 추출

```
SELECT SUBSTR('WELCOME TO ORACLE', 4, 3)
FROM dual;
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
W	E	L	C	O	M	E		T	O		O	R	A	C	L	E

④ 문제 : 'abcdefghijklmnopqrstuvwxy'에서 'efgh'만 추출하세요.

3) LENGTH : 문자열의 길이

```
SELECT LENGTH('WELCOME TO ORACLE')
FROM dual;
```

⑤ 문제 : '가나다라마바사아자차카타파하'의 길이를 구하세요.

4) INSTR : 명명된 문자의 위치를 구함

```
SELECT INSTR('WELCOME TO ORACLE', 'O', 3, 2)
FROM dual;
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
W	E	L	C	O	M	E		T	O		O	R	A	C	L	E

⑥ 문제 : 'WELCOME TO ORACLE'에서 두 번째 'L'의 위치를 구하세요.

5) LTRIM : 왼쪽 문자를 삭제

```
SELECT LTRIM('WELCOME TO ORACLE', 'W')
FROM dual;
```

6) RTRIM : 오른쪽 문자를 삭제

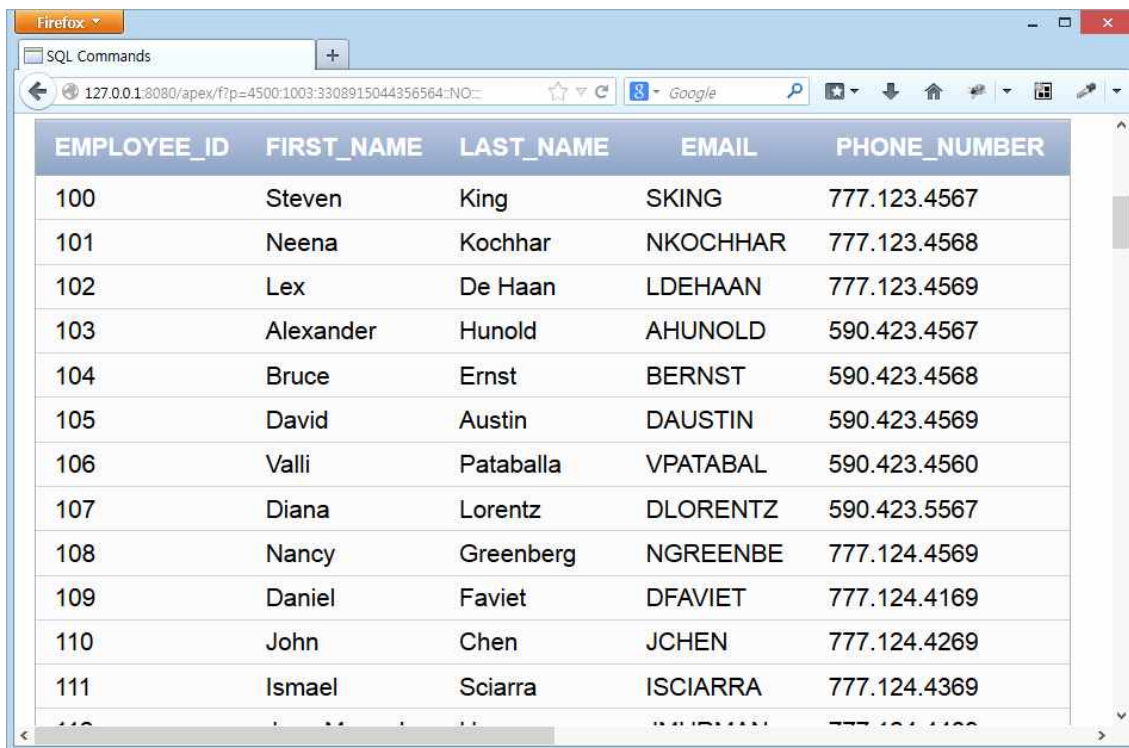
```
SELECT RTRIM('WELCOME TO ORACLE', 'E')  
FROM dual;
```

⑦ 문제 : 'abc'에서 'b'만 남은 문자열이 출력되게 하세요.

7) REPLACE : 특정 문자열을 대신 처리하는 함수

```
SELECT replace('WELCOME TO ORACLE', 'WELCOME TO', 'Hello')  
FROM dual;
```

⑧ 문제 : 515국이 모두 777로 변경되었다. 모두 바꿔 표현하세요.(아래쪽 생략)



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
100	Steven	King	SKING	777.123.4567
101	Neena	Kochhar	NKOCHHAR	777.123.4568
102	Lex	De Haan	LDEHAAN	777.123.4569
103	Alexander	Hunold	AHUNOLD	590.423.4567
104	Bruce	Ernst	BERNST	590.423.4568
105	David	Austin	DAUSTIN	590.423.4569
106	Valli	Pataballa	VPATABAL	590.423.4560
107	Diana	Lorentz	DLORENTZ	590.423.5567
108	Nancy	Greenberg	NGREENBE	777.124.4569
109	Daniel	Faviet	DFAVIET	777.124.4169
110	John	Chen	JCHEN	777.124.4269
111	Ismael	Sciarra	ISCIARRA	777.124.4369

5. 숫자형 함수

■ ROUND(반올림) 함수

```
SELECT ROUND(45.293, 2)  
FROM DUAL;
```

⑨ 문제 : 532.14에서 530이 되게 출력하세요.

■ TRUNC(버림) 함수

⑩ 문제 : 532.14에서 530이 되게 출력하세요.

■ MOD(나머지 구하는) 함수

⑪ 문제 : 사번이 홀수번인 사람들을 검색하세요.

■ POWER(거듭제곱) 함수

```
SELECT POWER(2,3)
FROM DUAL;
```

■ SQRT(제곱근) 함수

```
SELECT SQRT(4)
FROM DUAL;
```

■ CHR(아스키 코드에 해당되는 문자를 반환) 함수

아스키코드란?

아스키(ASCII) 또는 미국 정보 교환 표준 부호(American Standard Code for Information Interchange)는 영문 알파벳을 사용하는 대표적인 문자 인코딩이다. 아스키는 7비트 인코딩으로, 33개의 출력 불가능한 제어 문자들과 공백을 비롯한 95개의 출력 가능한 문자들로 이루어진다.

⑫ 문제 : ASCII 코드 '63'이 어떤 글자인지 출력해보세요.

6. 날짜 관련 함수

■ SYSDATE 함수

```
SELECT SYSDATE  
FROM DUAL;
```

■ 날짜 연산

$\text{SYSDATE} + 1$	내일 날짜를 구한다.
$\text{SYSDATE} - 1$	어제 날짜를 구한다.

⑬ 문제 : 오늘부터 100일 후가 며칠 인지 알아보세요.

⑭ 아래와 같이 각 사원들의 근무 일수를 구해보세요.(아래쪽 생략)

사번	이름	성	근무일수
102	Lex	De Haan	5171
206	William	Gietz	4661
204	Hermann	Baer	4661
203	Susan	Mavris	4661
205	Shelley	Higgins	4661
109	Daniel	Faviet	4591
108	Nancy	Greenberg	4590
114	Den	Raphaely	4478

■ MONTHS_BETWEEN : 두 날짜 사이의 월수를 계산

```
SELECT MONTHS_BETWEEN('2008/1/1', '1999/1/1')  
FROM dual;
```

⑯ 문제 : 각 사원들의 근무월수를 구하세요.

■ ADD_MONTHS : 해당 날짜에 월을 더한 날짜

```
SELECT ADD_MONTHS('2009/1/14', 6)
FROM DUAL;
```

■ LAST_DAY : 해당 월의 마지막 날짜

```
SELECT LAST_DAY('2009/1/14')
FROM DUAL;
```

⑰ 문제 : 2015년 5월이 며칠까지 있는지 알아보세요.

■ NEXT_DAY : 해당 날짜의 다음 요일의 날짜

```
NEXT_DAY('date', '요일')
```

⑱ 문제 : 2014년 3월의 첫 금요일이 며칠인지 알아보세요.

7. 형변환 함수

■ 문자열을 날짜 형으로 변환하는 TO_DATE 함수

```
TO_DATE('문자', '포맷')
```

■ 날짜나 숫자 형을 문자로 변환하는 TO_CHAR

```
TO_CHAR(숫자|날짜, '포맷')
```

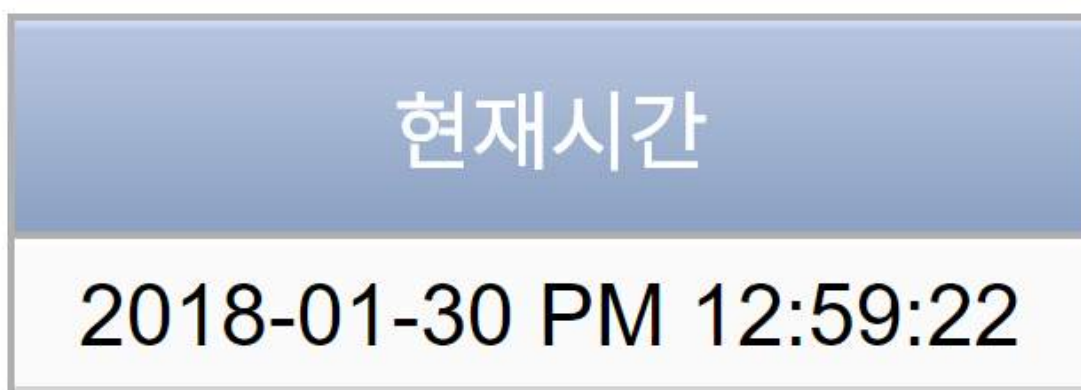
■ 날짜 형식 포맷

종 류	의 미
YYYY	년도 표현(4자리)
YY	년도 표현(2자리)
MM	월을 숫자로 표현
MON	월을 알파벳으로 표현
DD	일을 숫자로 표현
DAY	요일 표현
DY	요일을 약어로 표현

■ 시간 형식 포맷

종 류	의 미
AM or PM	오전(AM), 오후(PM) 시각 표시
HH or HH12	시간(1~12)
HH24	24시간으로 표현
MI	분 표현
SS	초 표현

①9 문제 : 현재 날짜와 시간을 초까지 표현하세요.



② 문제 : 입사일자를 '2001년 01월 13일' 형식으로 출력하세요.(아래쪽 생략)

사번	이름	성	입사일
100	Steven	King	2003년 06월 17일
101	Neena	Kochhar	2005년 09월 21일
102	Lex	De Haan	2001년 01월 13일
103	Alexander	Hunold	2006년 01월 03일
104	Bruce	Ernst	2007년 05월 21일
105	David	Austin	2005년 06월 25일
106	Valli	Pataballa	2006년 02월 05일
107	Diana	Lorentz	2007년 02월 07일

■ 숫자 표현 포맷

종 류	의 미	예	결과
9	한 자리의 숫자 표현	(1111, '99999')	1111
0	앞부분을 0으로 표현	(1111, '099999')	001111
\$	달러 기호를 앞에 표현	(1111, '\$99999')	\$1111
.	소수점을 표시	(1111, '99999.99')	1111.00
,	특정 위치에 , 표시	(1111, '99,999')	1,111
MI	오른쪽에 - 표시	(1111, '99999MI')	1111-
PR	음수값을 <>으로 표현	(1111, '99999PR')	<1111>
EEEE	과학적 표기법으로 표현	(1111, '99999EEEE')	1.111E+03
B	공백을 0으로 표현	(1111, 'B99999.99')	1111.00
L	지역 통화(local currency)	(1111, 'L99999')	\1111

② 문제 : 화폐기호 \$와 함께 3자리마다 콤마유형을 적용하여 급여를 출력하세요.

■ NVL

- null을 계산하면 값은 무조건 null
- null을 변경하기 위한 함수 nvl
- (상여금이 포함된) 실제봉급을 계산해봅시다.

```
select salary + salary * nvl(commission_pct,0)
from employees
order by 1 desc
```

■ DECODE

- 특정한 문자를 변경하고 싶을때

```
DECODE(값,비교1,변경값1,변경값2)
```

```
DECODE(값,비교1,변경값1,비교2,변경값2.....)
```

```
SELECT employee_id,first_name, last_name,
       decode(manager_id,null,'없음',manager_id) 매니저사번
from employees
```