

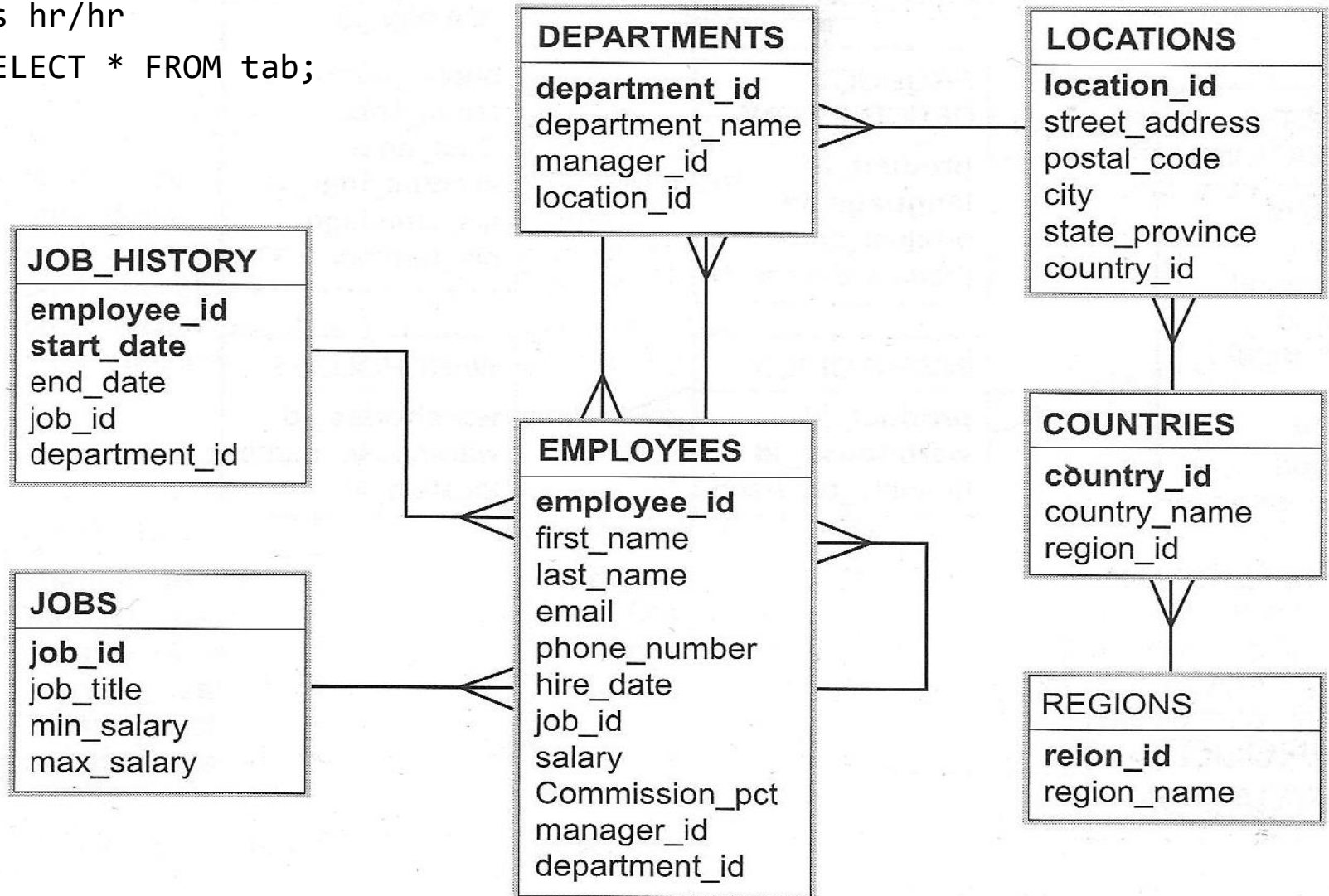
SQL의 기본

데이터 디렉터리 TAB

❖ HR01 소유하고 있는 테이블 조회

```
sqlplus hr/hr
```

```
SQL> SELECT * FROM tab;
```



테이블 구조를 살펴보기 위한 DESC

❖ DESC

- DESC[RIBE] 테이블명
- DESC dept;

```
SQL> desc departments;
```

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

테이블 구조를 살펴보기 위한 DESC

❖ NOT NULL 제약 조건

- NULL : 어떠한 값도 정해지지 않을 때 갖는 값
- NULL값을 저장하지 못하도록 하는 제약 조건

```
SQL> desc employees
```

Name	Null?	Type
-----	-----	-----
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
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)

테이블 구조를 살펴보기 위한 DESC

❖ 기본 키 제약 조건

- 테이블에서 각 로우를 유일하게 구분할 목적으로 사용하는 제약 조건
- 중복된 값을 가질 수 없다.
- NULL이 될 수 없다(반드시 값을 가져야 한다)

❖ 외래키 제약 조건

- 테이블 사이의 관계를 설정
 - DEPARTMENT_ID 부서 ID로 부서테이블(DEPARTMENTS) 참조
 - MANAGER_ID
- 다른 테이블에서 상세 정보가 저장되어 있고, 이에 참조 값을 가지는 것

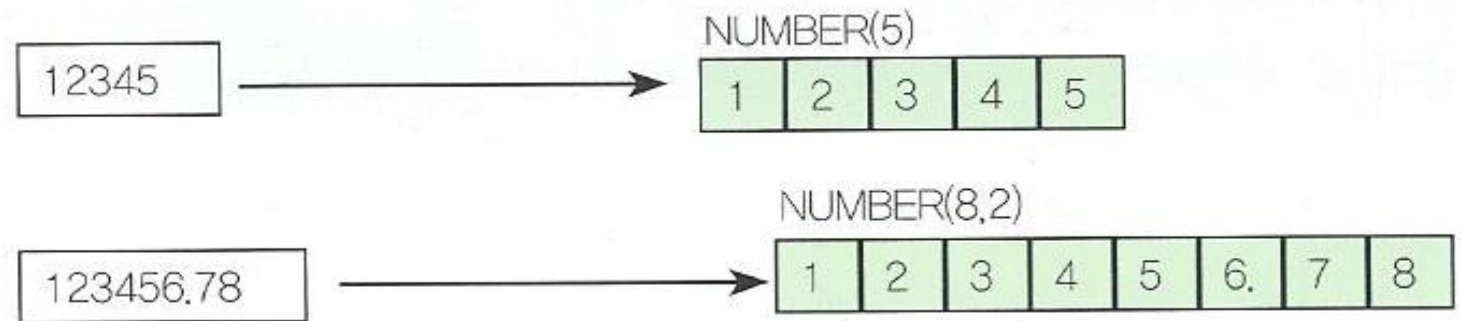
오라클의 데이터 형

❖ NUMBER

- NUMBER(precision, scale)
 - precision: 소수점을 포함한 전체 자릿수 지정
 - scale: 소수점 이하 자릿수 지정
 - scale 생략시 소수점 이하 반올림하여 정수 값만 저장

NUMBER(5)

NUMBER(8,2)



❖ DATE

- 날짜 타입
- 기본 형식 'YY/MM/DD'

오라클의 데이터 형

❖ CHAR

- 고정 길이 문자 데이터
- CHAR(20)

address

s	e	o	u	l															
---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

❖ VARCHAR2

- 가변길이 문자 데이터
- VARCHAR2(20)
 - 최대 20문자까지 가능하며 실제 문자열 길이만 큼 저장

address

s	e	o	u	l
---	---	---	---	---

데이터를 조회하기 위한 SELECT 문

❖ SELECT 문

○ 기본 형식

```
SELECT [DISTINCT] {*, column[Alias], ... }  
FROM table_name
```

```
SELECT *                → SELECT 절  
FROM departments; → FROM 절
```

```
SELECT employee_id, email  
FROM employees;
```

Cf] 라인당 출력될 문자 수 변경하기
- 디폴트는 80
set linesize 100

산술 연산자

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

종류	예
+	SELECT sal + comm FROM emp;
-	SELECT sal - 100 FROM emp;
*	SELECT sal * 12 FROM emp;
/	SELECT sal / 2 FROM emp;

NULL도 데이터!

❖ NULL의 특징

- 0이 아님
- 빈 공간이 아님
- 미확정(해당 사항 없음), 알 수 없는(Unknown) 값을 의미
- 어떤 값인지 알 수 없지만 값은 존재함
- ? 혹은 ∞ 의 의미
- 연산 할당, 비교가 불가능
 - $100 + ? = ?$
 - $100 + \infty = \infty$

NULL도 데이터!

❖ 커미션 계산 (NULL인 경우 ?)

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

❖ NULL을 0으로 처리하기

- NVL 함수 제공
- 커미션에 NULL이 저장되어 있더라도 NULL을 다른 값으로 변환하여 사용

```
SELECT last_name, salary, NVL(commission_pct,0), salary*12,  
       salary*12+salary*12*NVL(commission_pct,0)  
FROM employees;
```

컬럼 이름에 별칭 지정하기

❖ AS로 컬럼에 별칭 부여하기

```
SELECT last_name,  
       salary*12+salary*12*NVL(commission_pct,0) AS Annsal  
FROM employees;
```

❖ AS 없이 컬럼에 별칭 부여하기

```
SELECT last_name,  
       salary*12+salary*12*NVL(commission_pct,0) Annsal  
FROM employees;
```

❖ “” 로 별칭 부여하기

```
SELECT last_name,  
       salary*12+salary*12*NVL(commission_pct,0) "A n n s a l"  
FROM employees;
```

❖ 별칭으로 한글 사용하기

```
SELECT last_name,  
       salary*12+salary*12*NVL(commission_pct,0) "연봉"  
FROM employees;
```

Connection 연산자의 정의와 사용

❖ Connection 연산자

- 여러 개의 컬럼을 연결할 때 사용
- || 연산자

```
SELECT last_name, ' is a ', JOB_ID  
FROM employees;
```

→

Abel	is a	SA_REP
Ade	is a	SA_REP
Atkinson	is a	ST_CLERK

```
SELECT last_name || ' is a ' || JOB_ID  
FROM employees;
```

→

Abel is a SA_REP
Ade is a SA_REP
Atkinson is a ST_CLERK

DISTINCT 키워드

❖ DISTINCT

- 동일한 데이터 값들이 중복되어 출력되지 않도록 한번씩만 출력

```
SELECT department_id
FROM employees;
→
90
90
90
60
60
60
:
```

```
SELECT DISTINCT department_id
FROM employees;
→
100
30
(null)
90
20
70
110
50
80
40
60
10
```

실습

- ❖ **사원 테이블(EMPLOYEES)에 존재하는 직급의 종류를 출력하기 위해서 다음과 같은 쿼리문을 수행하면 중복된 내용이 출력 된다.**

```
SELECT job_id  
FROM employees;
```

다음과 같이 직급(JOB_ID)이 중복되지 않고 한 번씩 나열되도록 쿼리문을 작성한다

```
AC_ACCOUNT  
AC_MGR  
AD_ASST  
AD PRES  
AD_VP  
:
```