<SQL활용 서술형 시험 - 문제풀이>

[문항1] 다음 표를 기반으로 테이블을 생성하시오.

- 테이블명 : Title
- 테이블 생성 시 제약조건을 정의하시오.
- 제약조건 추가 시 Table level, Column level의 문법 중 자유롭게 사용하되 제약조건 이름을 생략하지 말고 고유하게 부여하시오.

Column명	TITLE_ID	TITLE	RATING	RELEASE_DATE
Key Type	PK			
Null/Unique		NN		
Check			G, PG, R	
Data type	NUMBER	VARCHAR2	VARCHAR2	DATE
Length	10	60	4	

정답:
```
create table title
(title_id number(10) constraint t_tid_pk primary key,
 title varchar2(60) constraint t_tit_nn not null,
 rating varchar2(4) constraint t_rat_ck check(rating IN ('G','PG','R')),
 release_date date);
```

[문항2] 알맞은 트랜잭션 제어 명령어를 작성하시오.

· 트랜잭션을 영구히 저장하는 명령어 – (① commit)

· 트랜잭션을 처음으로 되돌리는 명령어 – (② rollback)

※ 주어진 [상황]을 보고 <문항3> ~ <문항5>을 완성하시오.

[상황] ITWILL 교육센터에서 효율적인 학사관리 시스템을 개발하려고 한다. 귀하는 개발팀의 일원으로 우선적으로 수행해야 할 일은 데이터베이스에 관련된 문제점을 파악하는 것이다. 파악한 내용은 다음과 같다.

A. Student 테이블의 id 컬럼은 기본키(primary key)이고, name 컬럼 값은 입력하지 않는 경우 오류가 발생한다.
B. Student 테이블에서 ssn 컬럼의 값은 같은 행은 존재할 수 없고, major 컬럼은 Department 테이블의 기본키를 참조한다.
C. Department 테이블의 name 컬럼은 기본키이고, id 컬럼 값은 입력하지 않는 경우 오류가 발생한다.
D. Course 테이블의 instructor 컬럼은 Instructor 테이블의 기본키를 참조한다.
E. Instructor 테이블의 dept 컬럼은 Department 테이블의 기본키를 참조한다.

[문항3] STUDENT 테이블을 완성하시오.

```
SQL> CREATE TABLE Student (
        id      CHAR(8)     constraint s_id  ( ①  primary key  ),
        name    CHAR(13)    constraint s_name   NOT NULL,
        ssn     CHAR(10),
        major   CHAR(15),
        constraint s_aaa UNIQUE(  ② ssn ),
        constraint s_bbb FOREIGN KEY(major)
                        REFERENCES Department( ③  name  ) ) ;
```

[문항4] Department 테이블을 완성하시오.

```
SQL> CREATE TABLE Department (
        id      CHAR(5)  constraint d_id   ( ①  not null ),
        name    CHAR(15),
        constraint d_ccc PRIMARY KEY( ②  name  ) ) ;
```

[문항5] Course, Instructor 테이블을 완성하시오.

```
SQL>  CREATE TABLE Course (
        id      CHAR(8),
        name    CHAR(20)  constraint  c_name  NOT NULL,
        instructor  CHAR(5),
        constraint  c_id  PRIMARY KEY(id),
        constraint c_ddd FOREIGN KEY(instructor)
                        REFERENCE Instructor( ①  id  )) ;
SQL>  CREATE TABLE Instructor (
        id      CHAR(5),
        name    CHAR(15)  constraint  i_name  NOT NULL,
        dept    CHAR(15),
        constraint  i_id  PRIMARY KEY(id),
        contraint i_dept FOREIGN KEY( ② dept  )
                        REFERENCE Department( ③ name ) ) ;
```

[문항6] 트랜잭션 제어 명령어 중 Savepoint에 대해 설명하시오.

정답: 트랜잭션 진행 중 되돌아갈 지점(기점)을 지정하는 명령어

[문항7] sysdate를 다음과 같이 형식으로 출력하는 구문을 작성하시오.

```
TODAY
----------------------
2019-APR-05 Time 15:27
```

```
select to_char(sysdate, 'YYYY-MON-DD "Time" HH24:MI')
from dual;
```

[문항8] 다음 중 SQL 구문의 종류가 잘못 연결된 것은?

① DDL - CREATE, ALTER, DROP

② DML - INSERT, UPDATE, DELETE, TRUNCATE(DDL)

③ DCL - GRANT, REVOKE

④ DQL - SELECT

[문항9] 다음 SELECT 구문 중 성공적으로 실행되는 구문을 모두 고르시오.

정답 : 2, 3

```
1. SELECT first_name, last_name, job_id, salary*12
   AS Yearly Sal
   FROM   employees;
2. SELECT first_name, last_name, job_id, salary*12
   "yearly sal"
   FROM   employees;
3. SELECT first_name, last_name, job_id, salary AS
   "yearly sal"
   FROM   employees;
4. SELECT first_name||last_name AS name, job_id,
   salary*12 yearly sal
   FROM   employees;
```

[문항10] 다음 질문에 알맞은 키워드를 골라 작성하시오.

PRIMARY KEY	FOREIGN KEY	UNIQUE	CHECK
TABLE	VIEW	INDEX	SEQUENCE
SYNONYM	DESCRIBE	CASE	DECODE
SYSDATE	SELECT	FROM	WHERE
USER_TABLES	DISTINCT	GROUP BY	HAVING
NVL	NOT NULL	SUBSTR	ORDER BY

① 중복된 행을 자동으로 제거해 주는 키워드 DISTINCT

② 테이블 또는 뷰의 구조를 조회할 때 사용하는 명령어 DESCRIBE

③ 고유한 번호를 자동으로 생성해 주는 객체 SEQUENCE

④ 그룹을 제한하는 조건문을 작성할 수 있는 절 HAVING

⑤ 인수를 2개 받아들이는 함수로 NULL값을 실제 값으로 변환할 때 사용하는 함수 NVL

[문항11] EMPLOYEES 테이블로부터 department_id가 50인 사원들의 employee_id, last_name, salary값을 가지는 empvu50 뷰를 생성하는 DDL 구문을 작성하시오.

정답:
```
create view empvu50
    as select employee_id, last_name, salary
        from employees
        where department_id = 50;
```

[문항12] EMPLOYEES 테이블로부터 last_name이 'Abel'인 사원보다 급여를 더 많이 받는 사원의 employee_id, last_name, salary를 출력하는 SQL구문을 작성하시오.

정답:
```
select employee_id, last_name, salary
    from employees
    where salary > (select salary
                    from employees
                    where last_name = 'Abel');
```

[문항13] 부서별 사원의 수를 구하는 쿼리 구문을 작성하여 실행한 결과 다음과 같은 오류가 발생하였다. 정상적인 수행을 위해 쿼리구문을 재작성하시오.

```
SELECT department_id, COUNT(last_name)
FROM   employees;
```

ORA-00937: not a single-group group function
00937. 00000 - "not a single-group group function"

정답:
```
select ---
    from ---
    group by department_id;
```

[문항14] EMPLOYEES 테이블을 사용하여 사원의 LAST_NAME과 COMMISSION_PCT를 출력하되 커미션을 받는 사원의 정보만 출력하는 쿼리 구문을 작성하시오.

정답: select last_name, commission_pct
from employees
where commission_pct is not null;

[문항15] Employees테이블과 Employees테이블을 self-join하여, 사원들의 last_name(emp)과 사원의 매니저 last_name(mgr)을 함께 출력하는 쿼리 구문을 작성하시오. (단, 괄호 안의 이름을 column heading으로 출력하시오)

정답: select e1.last_name as "emp", e2.last_name as "mgr"
from employees e1 join employees e2
on e1.manager_id = e2.employee_id;

[문항16] EMP2 테이블을 다음과 같이 삭제하였다. 삭제된 테이블을 휴지통으로부터 되돌리는 구문을 작성하시오.

```
DROP TABLE emp2;
```

```
DROP TABLE emp2 succeeded.
```

정답: flashback table emp2 to before drop;

[문항17] 시퀀스 옵션 중 변경이 불가능한 옵션은?

① increment by 옵션

② start with 옵션

③ cycle 옵션

④ max/min 옵션

⑤ cache 옵션

[문항18] TITLE 테이블에 아래와 같이 데이터를 삽입하는 SQL(DML) 문장을 작성하시오.

TITLE_ID	TITLE	RATING	RELEASE_DATE
1	ORACLE	R	05-JUN-17

정답: insert into title
values (1, 'ORACLE', 'R', '05-JUN-17');

[문항19] EMPLOYEES 테이블로부터 last_name이 'A'로 시작되는 사원들의 employee_id, last_name, department_id를 출력하는 구문을 작성하시오.

정답: select employee_id, last_name, department_id
from employees
where last_name like 'A%';

[문항20] 다음 SQL 문장에서 틀린 라인의 번호를 적고 알맞게 수정하시오.

① SELECT department_id, AVG(salary) as avg_sal
② FROM employees
③ WHERE AVG(salary) > 6000
④ GROUP BY department_id;

정답: 번호 → ③
수정 → Having