

Homework

- **설명:** 데이터베이스 스키마를 생성하고 그 릴레이션(relation)에 튜플(tuple)들을 입력하십시오. 각 문제에 해당하는 SQL 쿼리(query)를 작성하십시오.
- **제출해야 하는 항목:**
 - **sql scripts:** 테이블 생성(table creation), 각 릴레이션에 튜플을 삽입(insertion), 그리고 질의하기(querying)를 포함한 과제에 관한 **모든 sql 명령들을 담고 있는 텍스트 파일**. 또한 아래의 설명과 같이 각 문제에 대해 **수행 결과들을 보여주는 sql 명령 (describe, select * ...)들을 포함**하십시오.
 - 각 문제의 파일 이름은 hw1.sql~hw8.sql로 한다. (hw문제번호.sql)
 - **doc (or hwp) report:** sql 쿼리(query)와 쿼리 결과를 보여주는 **실행 결과를 화면을 캡처**를 포함해야 합니다.
 - **문제 1의 테이블 만드는 명령어**들에서는 문제 1의 각 하위 문제를 수행한 후 **"describe tablename"**을 사용하여 만든 테이블 결과들을 표시하십시오
 - **2번 문제**의 insert 명령에서는 각각의 테이블의 내용을 볼 수 있게 **각각의 테이블의 "select *" 명령의 마지막 캡처 화면**(모든 데이터를 입력한 후에 마지막으로 한번 select 절을 수행)만 제출하십시오.
 - **문제 7과 8번**의 수정하는 쿼리(modification query)에서, **릴레이션이 수정되기 전과 후의 결과(select * ...)**를 보여주어야 한다.
- **제출 방법**
 - Online files: "학번.zip"의 이름으로 모든 파일들을 압축하여 하나의 파일로 사이버 강의실에 제출합니다.
- **1차 마감일: 2021년 4월 18일(일요일) 오후 11시**
 - 4월 18일(일요일) 오후 11시: 사이버 강의실
 - ◆ 범위: 과제 1번 ~ 4번
- **2차 마감일: 2021년 5월 2일(일요일) 오후 11시**
 - 5월 2일(일요일) 오후 11시: 사이버 강의실
 - ◆ 범위: 과제 5번 ~ 8번

● 평가와 벌점

- 매일 만점의 10%씩 감점, 마감일부터 5일이 지난 과제는 받지 않습니다.
- 잘못된 파일 제출에 대한 책임은 학생에게 있습니다. 사이버 강의실이나 출력물로 새로 제출한 과제가 마감 시간을 넘겼을 경우 늦게 제출 한 것으로 간주됩니다.
- 제출해야 할 항목들 중에 누락된 항목은 0점 처리됩니다. 예) sql script가 없는 것, 실행 결과를 캡처한 레포트가 없는 것, sql script의 출력물이 없는 것, 레포트의 출력물이 없는 것.
- 각각의 문제의 자세한 평가 기준은 교수와 수업 조교에 의해 정해집니다.

● 데이터베이스 스키마 설명

다음의 데이터베이스 스키마는 FIA 포뮬러 원 월드 챔피언십 (FIA Formula One World Championship)에 관한 것이다. 포뮬라 원은 운전석 하나에 바퀴가 겹으로 드러난 오픈휠 형식의 포뮬러 자동차 경주 중 가장 급이 높은 자동차 경주 대회이다. F-1 데이터베이스는 다음의 릴레이션을 포함한다. 각 속성(attribute)에 대한 적절한 데이터 타입을 사용하시오.

- Constructors (constructor, engine, country, races_entered, height, width)
- Drivers (name, birthday, country, constructor)
- Races (name, date, area)
- Results (race, driver, race_rank)

※ date, birthday 속성(attribute)는 **DATE** 타입이다.

Constructors 릴레이션은 constructor의 이름(name), 사용하는 엔진 종류(engine), constructor가 소속된 나라(country), 참가한 횟수(races_entered), 차체의 크기(height, width)를 기록한다.

Drivers 릴레이션은 드라이버의 이름(driver), 생일(birthday), 국적(country), 소속팀(constructor) 기록한다. **Races** 릴레이션은 경기 그랑프리의 이름(name)과 날짜(date)와 지역(area)을 저장한다. 마지막으로 **Results** 릴레이션은 각 그랑프리의 선수들의 성적을 저장한다.

각 선수는 같은 그랑프리에 두 번 이상 참가하지 않는다 race_rank는 각 선수가 참가한 그랑프리의 순위를 저장한다.

다음의 표들은 위의 릴레이션들의 샘플 데이터이다. 모든 드라이버들은 **Constructors** 릴레이션에 포함되고 경기에 참가한 드라이버들은 **Results** 릴레이션에 기록된다.

constructor	country	engine	racess_entered	height	width
McLaren	British	Mercedes	884	95	180
BMW Sauber	German	BMW	70	95	180
Renault	French	Renault	660	91	175
Ferrari	Italian	Ferrari	1010	93	180
Toyota	Japanese	Toyota	1140	91	175
Red Bull	Austrian	Honda	304	91	175
Honda	Japanese	Honda	88	91	175
Toro Rosso	Italian	Ferrari	268	93	180
Jaguar	British	Cosworth	85	93	178

(a) Sample data for relation **Constructors**

driver	birthday	country	constructor
hamilton	07/01/1985	British	McLaren
heidfeld	10/05/1977	German	BMW Sauber
alonso	29/07/1981	Spanish	Renault
raikkonen	17/10/1979	Finnish	Ferrari
kubica	07/12/1984	Polish	BMW Sauber
massa	25/04/1981	Brazilian	Ferrari
trulli	13/07/1974	Italian	Toyota
webber	27/08/1976	Australian	Red Bull
button	19/01/1980	British	Honda
vettel	03/07/1987	German	Toro Rosso
barrichello	23/05/1972	Brazilian	Honda
Irvine	10/11/1965	British	Jaguar
Schmacher	03/01/1969	German	Ferrari

(b) Sample data for relation **Drivers**

Name	beginDate	area
Australian Grand Prix	2009-03-29	Australia
British Grand Prix	2009-06-21	Europe
German Grand Prix	2009-07-12	Europe
European Grand Prix	2009-08-23	Europe
Abu Dhabi Grand Prix	2009-11-01	Middle East,
Malaysian Grand Prix	2008-03-23	Asia
Spanish Grand Prix	2008-04-27	Europe

Singapore Grand Prix	2008-09-28	Asia
Brazilian Grand Prix	2008-11-02	South America
Monaco Grand Prix	2007-05-27	Africa
Canadian Grand Prix	2007-06-10	North America
United States Grand Prix	2007-06-17	North America
French Grand Prix	2007-07-01	Europe
Italian Grand Prix	2007-09-09	Europe
San Marino Grand Prix	2004-04-25	Europe

(c) Sample data for relation **Races**

Race	driver	race_rank
Australian Grand Prix	trulli	first place
Australian Grand Prix	barrichello	second place
Australian Grand Prix	button	third place
British Grand Prix	button	first place
British Grand Prix	barrichello	second place
British Grand Prix	vettel	third place
German Grand Prix	button	first place
German Grand Prix	vettel	second place
German Grand Prix	webber	third place
European Grand Prix	button	first place
European Grand Prix	barrichello	second place
European Grand Prix	webber	third place
Abu Dhabi Grand Prix	button	first place
Abu Dhabi Grand Prix	vettel	second place
Abu Dhabi Grand Prix	barrichello	third place
Malaysian Grand Prix	hamilton	first place
Malaysian Grand Prix	heidfeld	second place
Malaysian Grand Prix	raikkonen	third place
Spanish Grand Prix	hamilton	second place
Singapore Grand Prix	hamilton	first place
Singapore Grand Prix	massa	second place
Singapore Grand Prix	kubica	third place
Brazilian Grand Prix	hamilton	first place
Brazilian Grand Prix	massa	second place
Monaco Grand Prix	alonso	first place
Monaco Grand Prix	hamilton	second place

Monaco Grand Prix	massa	third place
Canadian Grand Prix	hamilton	first place
Canadian Grand Prix	alonso	second place
Canadian Grand Prix	massa	third place
United States Grand Prix	hamilton	first place
United States Grand Prix	alonso	second place
United States Grand Prix	massa	third place
French Grand Prix	hamilton	first place
French Grand Prix	alonso	second place
French Grand Prix	massa	third place
Italian Grand Prix	hamilton	first place
Italian Grand Prix	alonso	second place
San Marino Grand Prix	Schmacher	first place
San Marino Grand Prix	barrichello	second place

(d) Sample data for relation **Results**

● **Problems**

- 위에서 보여준 약식 스키마와 샘플 데이터를 기반으로 **테이블 만드는 선언**(table creation declaration)을 SQL로 작성하라. 또한 만드는 릴레이션에는 기본키(Primary key)와 외래키(foreign key)들이 기술되어야 한다. 1-a)부터 1-d)까지 문제들을 수행한 후에는 생성된 테이블 결과를 보여주기 위해 "describe tablemane"을 사용하시오.

- Constructors 릴레이션에 적합한 스키마.
- Drivers 릴레이션에 적합한 스키마.
 - Birthday 속성(attribute)은 DATE 타입을 사용하시오.
- Races 릴레이션에 적합한 스키마.
 - date 속성(attribute)은 DATE 타입을 사용하시오.
- Results 릴레이션에 적합한 스키마.

※ 새로운 속성(attribute)을 추가하지 않고 위에서 설명한 속성 중에서 릴레이션의 기본키를 정한다.

- 위의 샘플들에서 보여준 콘텐츠(contents)를 각 릴레이션의 콘텐츠로 생성하기 위해 다음의 **insert SQL 쿼리**를 작성하시오. 외래키(foreign key) 관계를 바탕으로 Constructors, Drivers, Races, Results 테이블로 데이터를 **삽입하는 정확한 순서**를 고려하라. 2-a)부터 2-d)까지 문제들을 수행한 후에는 **각 테이블의 결과에 'select *'을 실행하여 모든 삽입의 마지막 결과를** 보여주시오.

- a) 첫 번째 릴레이션을 데이터 삽입
- b) 두 번째 릴레이션을 데이터 삽입
- c) 세 번째 릴레이션을 데이터 삽입
- d) 네 번째 릴레이션을 데이터 삽입

3. 위의 데이터베이스 스키마를 기반으로 하나의 릴레이션에 관한 다음의 간단한 SQL 쿼리 (simple SQL query)를 작성하시오

- a) 그랑프리에 100번 이상 참가한 constructor의 이름과 constructor의 국가(country)를 찾아라.
- b) 차체의 높이가 95센치가 아니고 차체의 너비가 180 센티보다 작은 차체를 사용하는 레이싱 팀의 이름, 엔진이름, 그랑프리 참가 횟수를 찾아라. 결과는 첫번째는 엔진이름으로, 두 번째는 그랑프리에 참가한 횟수를 기준으로 오름차순으로 보여준다.
- c) Ferrari 엔진을 사용하는 모든 constructor들을 찾아라
- d) 1980년도 이후에 태어난 모든 드라이버의 이름을 찾아라. 단, 결과 테이블의 컬럼은 새 이름으로 newBoys을 사용한다.
- e) 각 그랑프리에서 우승한 드라이버의 이름과 그랑프리의 이름을 찾아라.
- f) 유럽 지역에서 열린 그랑프리들 모두 찾아라. 그랑프리 이름을 기준으로 오름차순으로 정렬하여라
- g) 문자 'H'로 시작하는 이름을 가진 모든 드라이버를 찾아라.
- h) 두 개 이상의 단어로 이루어진 이름을 가지는 모든 constructor들의 이름을 찾아라.
예) Red Bull

4. 위의 데이터베이스 스키마를 기반으로 다음의 SQL 쿼리를 작성하라

- a) 이탈리아 국적의 레이싱 팀 소속의 드라이버들을 찾아라.
- b) 소속 레이싱 팀의 국적과 드라이버의 국적이 같은 드라이버를 찾아라.
- c) Monaco Grand Prix 그랑프리에 참가한 드라이버의 이름, 드라이버가 소속된 레이싱 팀의 엔진과 그랑프리 참가 횟수를 열거하여라.
- d) 오직 하나의 드라이버를 가지는 모든 레이싱 팀을 찾아라.
- e) BMW Sauber 레이싱 팀의 멤버이거나 Spanish Grand Prix 그랑프리에 참가한 모든 드라이버의 이름을 중복없이 열거하여라.
- f) 드라이버의 이름과 드라이버가 소속된 레이싱 팀의 참가 횟수를 내림차순으로 열거하시오. (첫 번째 정렬기준은 그랑프리 참가 횟수, 두 번째 정렬기준은 드라이버의 이름이다.)
- g) Cosworth 엔진을 사용하는 레이싱 팀과 Mercedes 엔진을 사용하는 레이싱 팀을 모두 가지고 있는 모든 나라를 찾아라
- h) 그랑프리에서 우승한 적이 있으면서 순위에는 들었으나 우승을 하지 못했던 드라이버들을 중복없이 찾아라.

- i) 2008년부터 2009년까지 2년 동안 개최한 그랑프리에 참가한 드라이버의 이름과 드라이버가 참가한 그랑프리의 **날짜(begindate)**의 **월과 년도**(형식: mm-yyyy)를 열거하라.
5. 위의 데이터베이스 스키마를 기반으로 **subquery**를 포함하는 다음의 SQL 쿼리를 작성하라. 각 문제의 답에 **적어도 하나 이상의 subquery를 사용하여야 하고 각 문제의 쿼리는 다른 문제의 쿼리와 다른 방법으로 작성하시오.** (예. 각 방법은 EIXSTS, IN, ALL, SOME 그리고 ANY와 같은 다른 집합 연산자들을 사용해야 하고 NOT을 같이 사용할 수도 있다)
- 그랑프리에 가장 많이 참가한 레이싱 팀을 소유한 나라(country)를 찾아라.
 - 그랑프리에서 적어도 한번은 우승을 한 드라이버가 소속된 레이싱 팀을 찾아라.
 - Mercedes 엔진을 사용하는 드라이버들의 이름을 찾아라.
 - Ferrari 레이싱 팀의 드라이버들이 참가한 그랑프리를 찾아라.
 - 차체의 높이는 같으나 다른 팀의 차체보다 너비가 넓은 차체를 사용하는 드라이버들의 이름을 찾아라
6. 위의 데이터베이스 스키마를 기반으로 **집계 함수(aggregate function)**를 포함하는 다음의 SQL 쿼리를 작성하시오
- Ferrari 엔진을 사용하는 레이싱 팀의 개수를 구하라.
 - 영국 국적의 레이싱 팀들의 그랑프리 참가 횟수의 평균을 구하여라.
 - 각 레이싱 팀에 대해 각 팀에서 가장 나이가 어린 드라이버의 생일을 구하여라.
 - 독일 출신의 드라이버가 속한 레이싱 팀의 차체 높이의 평균을 구하라.
 - 그랑프리에서 우승을 한 적이 있는 레이싱 팀에 대해 각 레이싱 팀의 우승한 횟수를 구하여라.
 - 적어도 2 명의 드라이버를 가지고 있으면서 우승을 한 적이 있는 레이싱 팀의 수를 구하여라.
 - 각 국가 별, 레이싱 팀에 속한 모든 드라이버의 평균 나이를 구하라. 한 국가에 속한 레이싱 팀이 하나 이상일 경우, 레이싱 팀별로 표시한다.
hint) TIMESTAMPDIFF 함수 사용
 - 각 그랑프리 별, 두 명 이상의 선수가 순위에 든 레이싱 팀을 구하라.
7. 위의 데이터베이스 스키마를 기반으로 수정(modification)을 포함하는 다음의 SQL 쿼리를 작성하시오. 수정된 릴레이션의 **릴레이션이 수정되기 전과 후의 결과(select * ...)**를 보여라. 스크립트는 릴레이션 수정 전의 결과를 보여야 하고 문제의 SQL을 실행시키고, 릴레이션 수정 후의 결과를 보여야 한다. 각 하위문제는 하나씩 수행되어야 한다.(즉, 각 하위문제를 수행한 후에 데이터베이스를 초기화할 필요가 없다.)
- BMW Sauber 레이싱 팀에 두 명의 드라이버 Russell와 Villeneuve를 영입하였다. 영국 출신(British)의 Russell의 생년월일은 15/02/1998이다. 캐나다 출신(Canadian)의

Villeneuve의 생년월일은 09/04/1971이다. 이러한 사실을 데이터베이스에 삽입하라.

- b) 스위스(Swiss)에서 새로운 레이싱 팀 Sauber가 만들고 독일 출신의 Frentzen를 영입했다. Frentzen은 1967년 5월 18일이 생일이며 Mercedes 엔진을 사용하고 차체의 높이와 너비는 각각 93센티와 180 센티이다. 이러한 사실을 데이터베이스에 삽입하라.

- c) 아시아 지역의 그랑프리에서 3등을 차지한 모든 드라이버들을 Drivers 릴레이션에서 삭제하라. 해당 드라이버의 경기 기록도 삭제한다.

Hint) 데이터를 저장하기 위한 새로운 테이블을 만들 수 있다.

- d) 차체의 높이와 너비를 인치(one inch = 2.5 centimeter)로 측정되도록 Constructors 릴레이션을 수정하라.

- e) Honda 엔진을 사용하는 모든 드라이버는 그랑프리에서 기권을 한다. Results 릴레이션을 수정하라.

※ 기권은 race_rank를 'drop out'로 표현한다.

- f) 드라이버를 두 명 이상 가지고 있는 모든 레이싱 팀의 드라이버를 삭제하라. 해당 드라이버의 경기 기록도 삭제한다.

Hint) 데이터를 저장하기 위한 새로운 테이블을 만들 수 있다.

- 8. 위의 데이터베이스 스키마를 기반으로 스키마 수정(modification)을 관련된 다음의 SQL 쿼리를 작성하시오

- e) 1-(a)의 Constructors 릴레이션의 스키마로부터 races_entered 속성(attribute)을 삭제된 Constructors 릴레이션으로 수정(alteration)하라.

- a) 1-(b)의 Constructors 릴레이션의 스키마로부터 tire 속성(attribute)을 포함하는 수정된 Constructors 릴레이션으로 수정(alteration)하라.

※ tire 속성의 값은 KoreanCompany이다.