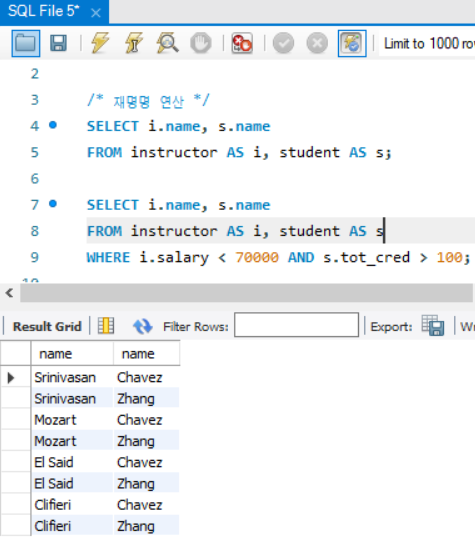
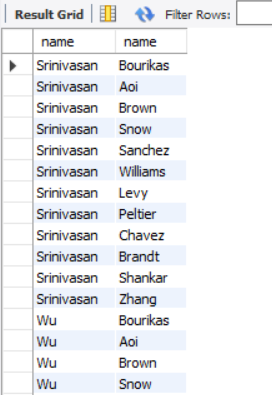
**3주차 실습과제**

**20175105\_곽영주**

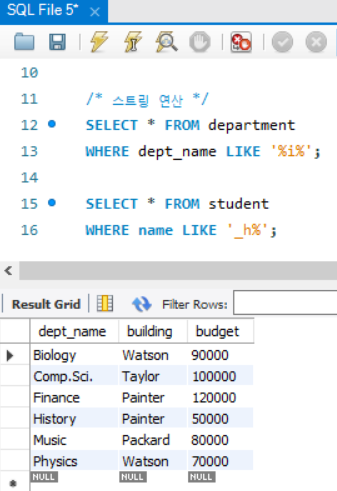
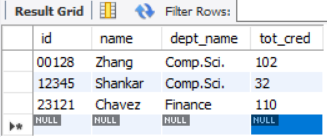
**과제 1-1**

1. **재명명 연산**

instructor와 student의 테이블 명을 i와 s로 재명명하여 테이블 명을 간단히 사용할 수 있게 만들고 AND 연산으로 통해 instructor의 salary < 70000와 student의 tot\_cred > 100이란 두 조건이 다 만족하는 데이터 결과를 보여줍니다.

1. **스트링 연산**

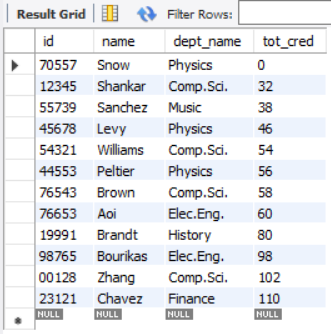
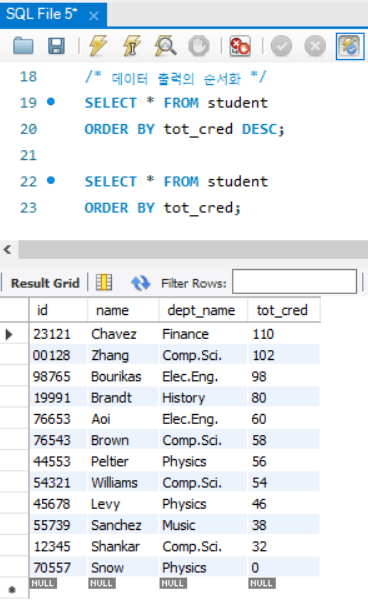
 

%는 모든 문자를 의미하고 \_는 하나의 문자만 의미합니다.

‘%i%’는 앞뒤로 문자 개수 상관없이 i를 포함하고 있는 데이터 결과를 보여줍니다.

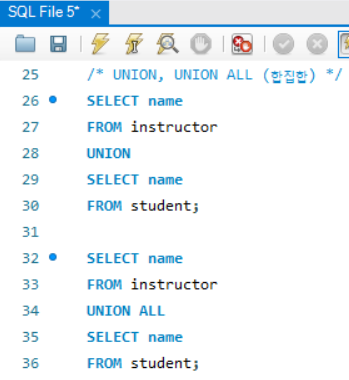
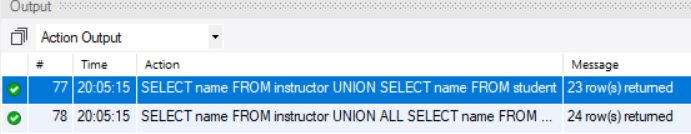
‘\_h%’는 뒤는 문자 개수 상관없고 앞에서 두 번째에 h를 포함하고 있는 데이터 결과를 보여줍니다.

1. **데이터 출력의 순서화**



ORDER BY는 데이터 출력의 순서화를 하는 것이고 기본값은 ASC(오름차순)이고 DESC라는 키워드를 사용하면 내림차순입니다.

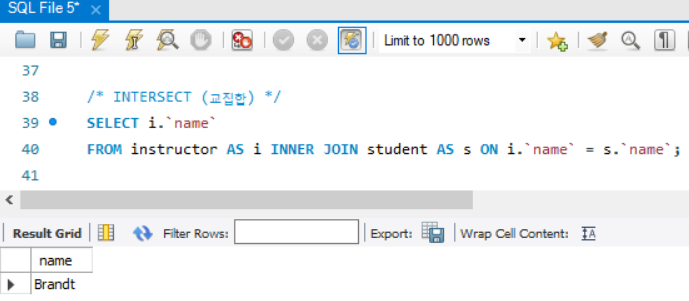
1. **UNION, UNION ALL (합집합)**

UNION은 중복을 제거한 데이터 결과를 보여줍니다.

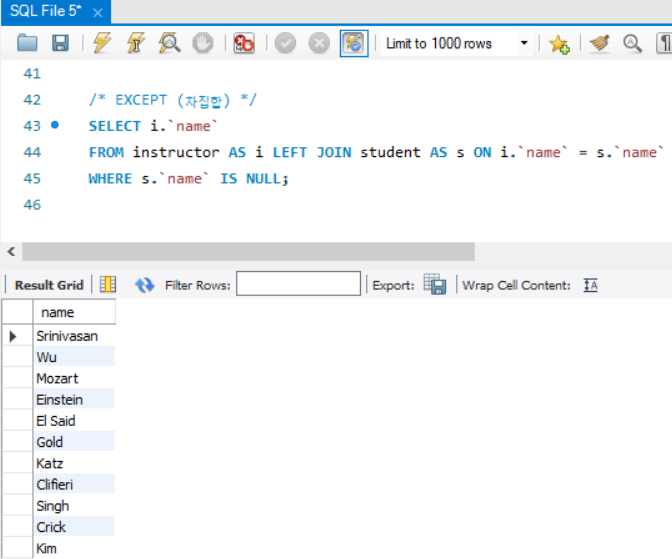
UNION ALL은 중복을 제거하지 않고 데이터 결과를 보여줍니다.

1. **INTERSECT (교집합)**



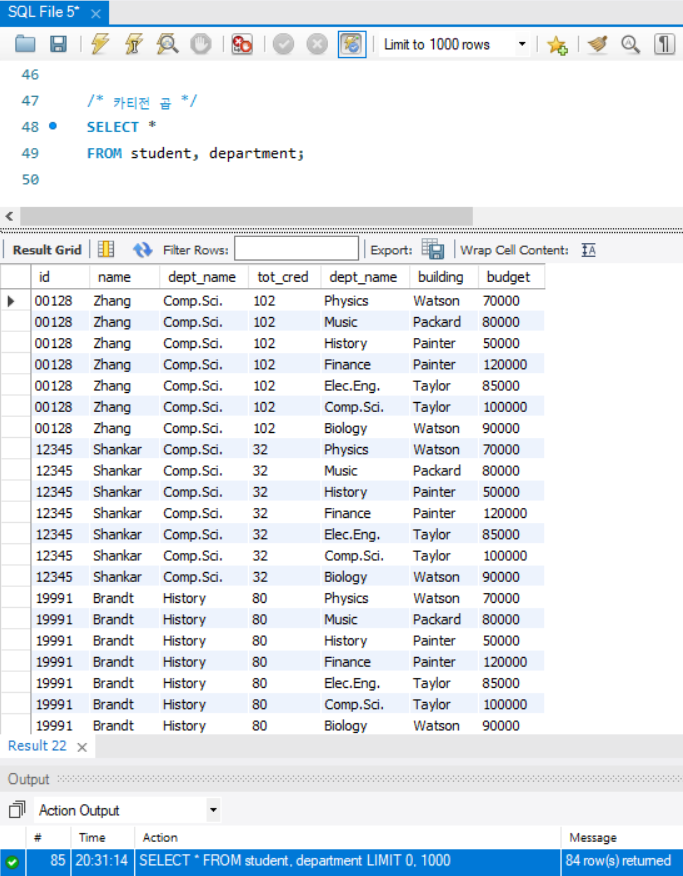
Instructor와 student를 조인한 결과에서 instructor.name과 student.name이 같은 것을 보여줍니다. 즉, 교집합의 결과를 보여줍니다.

1. **EXCEPT (차집합)**



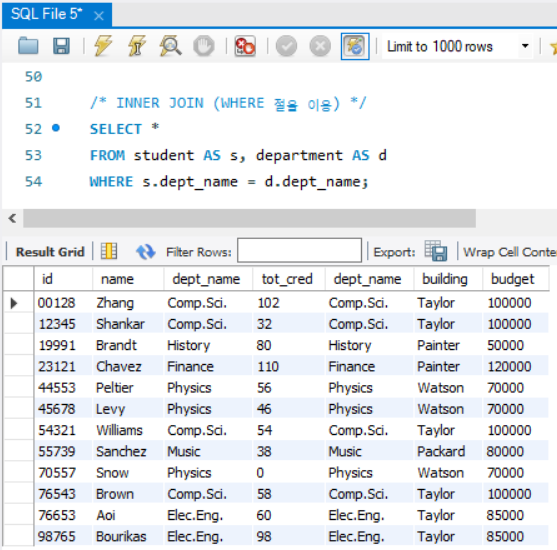
Left 조인을 사용하여 instructor기준으로 조인된 데이터들 중에 student.name이 NULL인 경우를 보여줍니다. 즉, 차집합의 결과를 보여줍니다.

1. **카티전 곱**



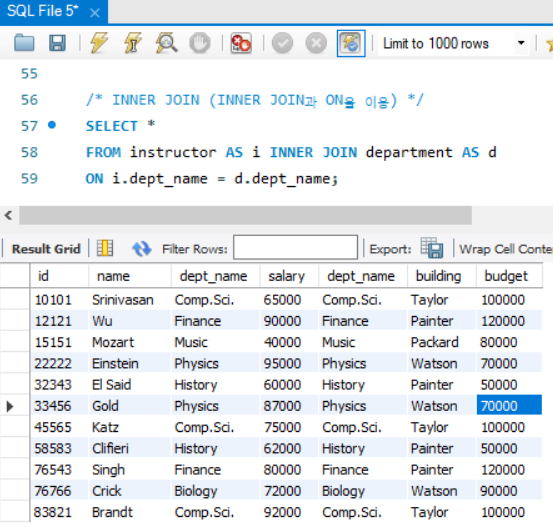
카티전 곱은 두 데이터에 대해서 가능한 모두의 경우의 수를 계산해서 보여줍니다.

1. **INNER JOIN (WHERE 절을 이용)**



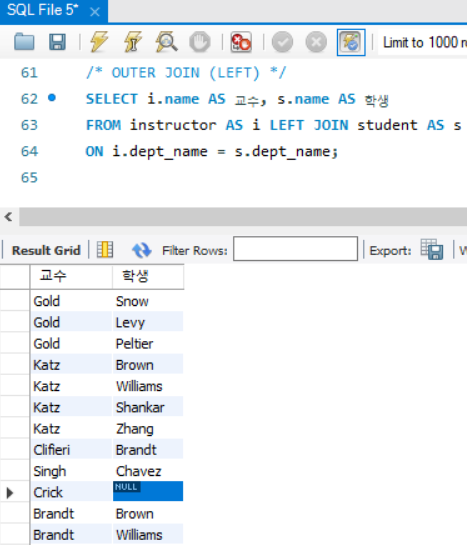
dept\_name이라는 공통 컬럼의 값이 같은 데이터만 매치하여 공통 컬럼을 포함하는 데이터 집합의 결과를 보여줍니다.

1. **INNER JOIN (INNER JOIN과 ON을 이용)**



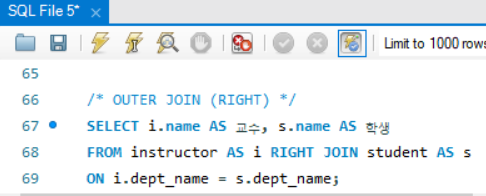
INNER JOIN과 ON을 사용하여 dept\_name이라는 공통 컬럼의 값이 같은 데이터만 매치하여 공통 컬럼을 포함하는 데이터 집합의 결과를 보여줍니다.

1. **OUTER JOIN (LEFT)**



LEFT 조인을 사용하여 instructor가 기준이 될 때 student에는Crick 데이터에 해당하는 dept\_name이 존재하지 않는 것을 보여줍니다.

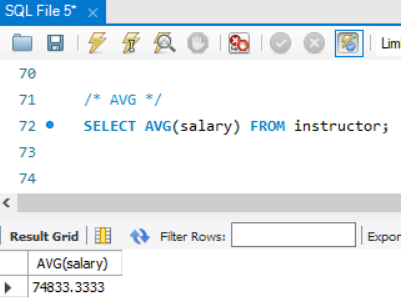
1. **OUTER JOIN (RIGHT)**

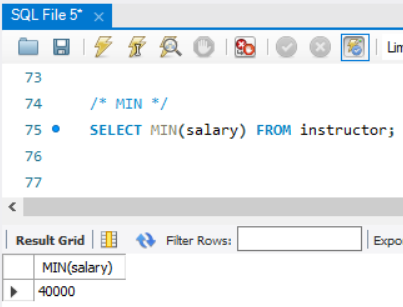
LEFT 조인에서 있던 NULL값이 존재하지 않는 이유는 RIGHT 조인을 사용하여 student에 기준을 뒀기 때문에 NULL은 뺀 나머지 데이터들을 보여주기 때문입니다.

**과제 1-2**

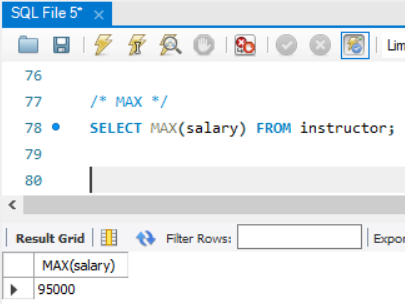
1. **집성함수 (AVG, MIN, MAX, SUM, COUNT)**



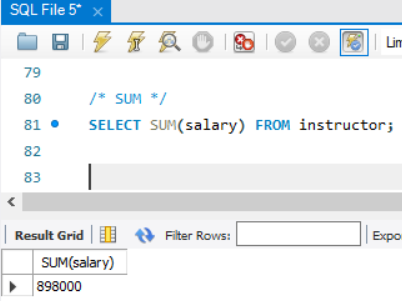
instructor.salary 데이터들의 평균값을 보여줍니다.



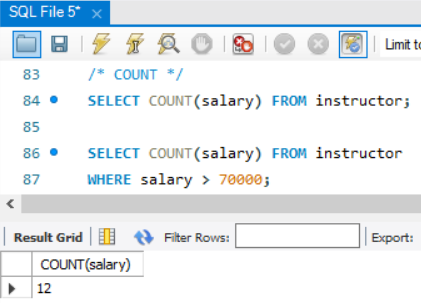
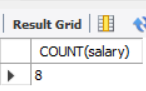
instructor.salary 데이터들 중 최소값을 보여줍니다.



instructor.salary 데이터들 중 최대값을 보여줍니다.



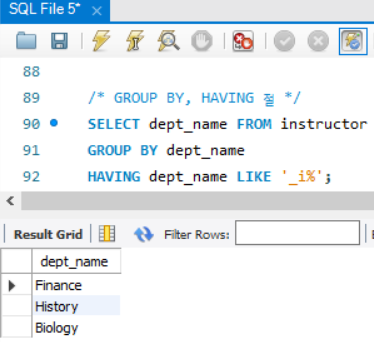
instructor.salary 데이터들의 합을 보여줍니다.

instructor.salary 데이터들의 개수를 보여줍니다.

instructor.salary 데이터들 중 70000보다 큰 instructor.salary 데이터 개수를 보여줍니다.

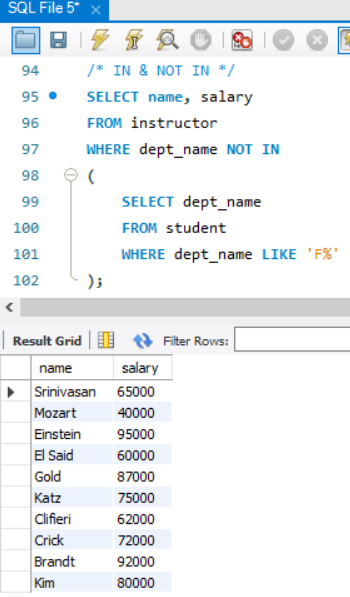
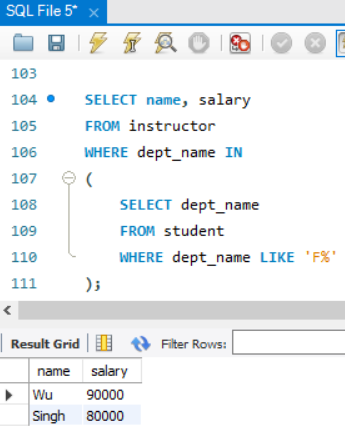
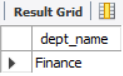
1. **GROUP BY, HAVING 절**



HAVING절은 GROUP BY를 통해 그룹 지어진 데이터에 대한 조건을 부여합니다.

‘\_i%’라는 조건을 부여하여 두 번째 문자가 i로 시작하는 데이터 그룹을 보여줍니다.

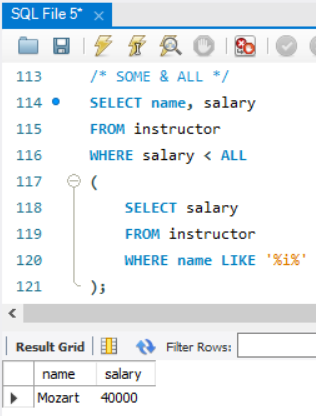
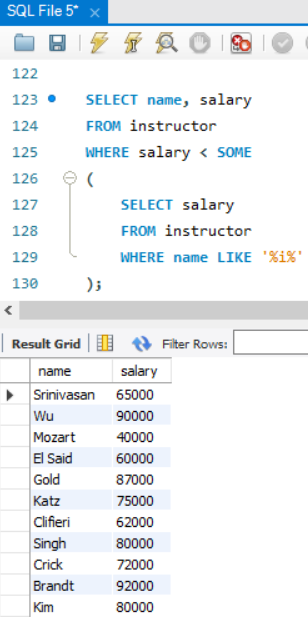
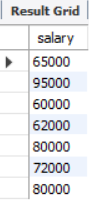
1. **IN & NOT IN**

IN : student.dept\_name 데이터 중 F로 시작하는 dept\_name 데이터와 같은 instructor.dept\_name을 포함하는 name, salary결과를 보여줍니다.

NOT IN : student.dept\_name 데이터 중 F로 시작하는 dept\_name 데이터와 같은 instructor.dept\_name을 포함하지 않는 name, salary결과를 보여줍니다.

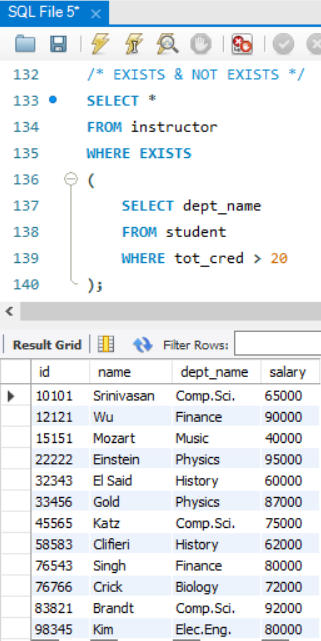
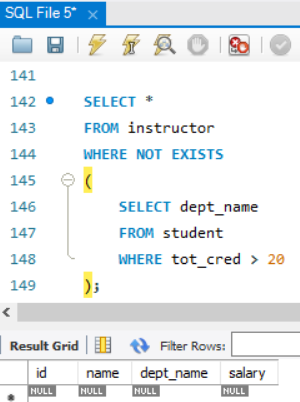
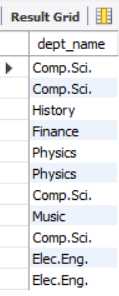
1. **SOME & ALL**

SOME : instructor.salary 데이터들 중에 i를 포함하는 데이터를 추출한 데이터들 중에 하나라도 만족하는 instructor의 name, salary의 데이터만 보여줍니다.

ALL : SOME과 반대로 부질의 부분에서 추출한 데이터들을 모두 만족하는 instructor의 name, salary의 데이터만 보여줍니다.

1. **EXISTS & NOT EXISTS**

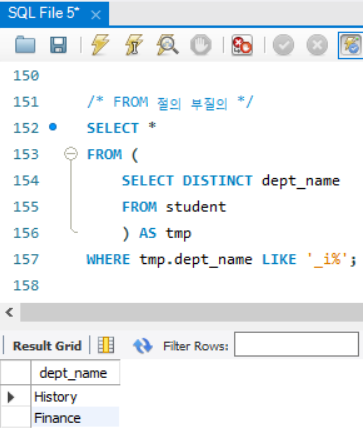
  

EXISTS는 빈 테이블이 있는지 없는지 검사하는 연산자입니다.

중첩 부질의 부분에 값이 존재하기 때문에 EXISTS는 TRUE이므로 해당되는 instructor 데이터를 보여줍니다.

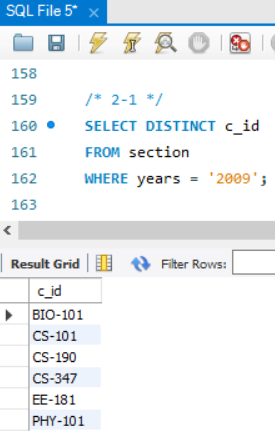
하지만 NOT EXISTS는 TRUE인 데이터에 NOT을 붙였기 때문에 FALSE이므로 빈 데이터를 보여줍니다.

1. **FROM 절의 부질의**

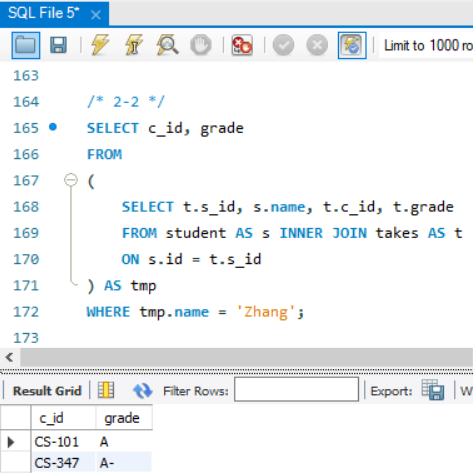


FROM절의 부질의 부분에서 student.dept\_name 데이터들을 DISTINCT를 통해 중복 제거를 해주고 FROM절의 부질의에서 추출한 데이터에서 두 번째 문자가 i로 시작하는 데이터들을 보여줍니다.

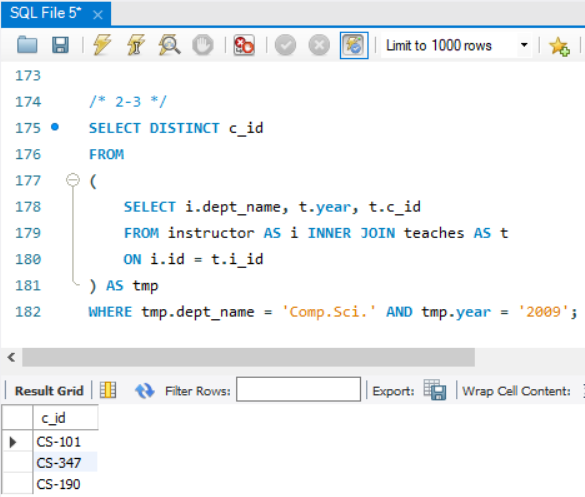
**과제 2**



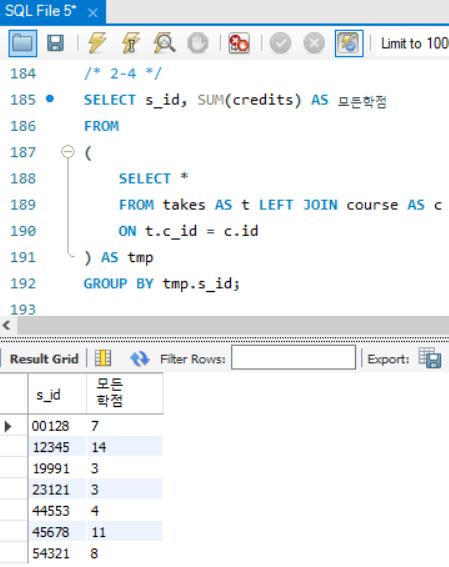
Section 테이블에서 2009년도 연 모든 과목의 코드를 DISTINCT를 사용하여 중복 없이 검색했습니다.



FROM절의 부질의를 부분에서 student와 takes의 테이블을 학생 코드로 s\_id, name, c\_id, grade 데이터를 INNER JOIN하고 그 데이터 집합에서 ‘Zhang’ 이라는 학생이 들었던 과목의 코드와 성적을 검색했습니다.



FROM절의 부질의를 부분에서 instructor와 teaches의 테이블을 강사 코드로 dept\_name, years, c\_id 데이터를 INNER JOIN하고 그 데이터 집합에서 AND 연산을 통해 ‘Comp.Sci.’ 학과에 속한 강사가 진행한 수업 중 2009년에 수업한 과목의 코드를 DISTINCT를 사용하여 중복 제거하고 검색했습니다.



FROM절의 부질의 부분에서 takes 테이블 기준으로 course 테이블을 과목 코드로 데이터를 LEFT JOIN하고 GROUP BY를 통해 학생 코드로 그룹 짓고 SUM 연산자를 통해 그룹 지어진 학생 코드에 대한 credits 값들을 더해 총 학점 수(누적된 학점 수)를 구했습니다.