

**CS 386/586 Winter 2012**  
**Assignment 2**  
**Suggested Answers**

Part I Extensions to the SELECT clause in SQL

1. (4 points) Write an SQL query that lists the agent\_id and salary – in Euros based on a conversion rate available for the day that you do your assignment. (You can find a conversion rate online.) Set the name of the attribute in the final query answer (for the salary in Euros) to be ‘Salary-in-Euros’. Show the SQL query, (at most) the first five rows of your query answer, and the number of rows in the query answer.

```
SELECT agent_id, (salary * 0.77) AS "Salary-in-Euros"  
FROM Agent
```

agent_id	Salary-in-Euros
1	38925.81
2	39235.35
3	43058.40
4	51246.58
5	58824.92

662 rows

2. (4 points) Write an SQL query that finds the minimum, maximum, and average salaries for agents that are from Germany. Choose appropriate attribute names for all columns of your query answer. Show the SQL query, (at most) the first five rows of your query answer, and the number of rows in the query answer.

```
SELECT MIN (salary) AS min_sal,  
       MAX (salary) AS max_sal,  
       AVG (salary) AS avg_sal  
FROM Agent  
WHERE country = 'Germany'
```

min_sal	max_sal	avg_sal
50177	78903	67159.000000000000

1 row

## Part II Relational Algebra (select, project, join, cross product) and SQL

For questions 3, 4, and 5 write a query in relational algebra, write and execute an equivalent query in SQL against the introdb\_spy database. Turn in: (a) your relational algebra query, (b) your SQL query, and (c) (at most) the first five rows of your query answer along with the count of rows in your query answer.

3. (7 points) List the agent\_id, first, middle, and last names of agents with the team\_id and team name for the teams that they are assigned to.

$\pi_{A.agent\_id, A.first, A.middle, A.last, T.team\_id, T.name} (Agent \bowtie_{A.agent\_id = TR.agent\_id} TeamRel \bowtie_{TR.team\_id = T.team\_id} Team)$

```
SELECT A.agent_id, A.first, A.middle, A.last, T.team_id, T.name
FROM Agent A NATURAL JOIN TeamRel TR NATURAL JOIN Team T
```

agent_id	first	middle	last	team_id	name
168	Richard	NULL	Venkatesh	1	Renegade
488	Bill	NULL	White	1	Renegade
560	George	NULL	Flood	1	Renegade
581	Nicholas	NULL	Larsen	1	Renegade
604	Crispin	NULL	Rios	1	Renegade

374 rows

(4 points) Write the relational algebra query a second time. (If your first relational algebra query used the join operator, then the second relational algebra query should NOT use the join operator. Conversely, if your first relational algebra query did not use the join operator, then your second relational algebra query SHOULD use the join operator.) Turn in the relational algebra query.

$\pi_{A.agent\_id, A.first, A.middle, A.last, T.team\_id, T.name} (\sigma_{A.agent\_id = TR.agent\_id \text{ AND } TR.team\_id = T.team\_id} (Agent \times TeamRel \times Team))$

Another way to write this query without using the join operator is as shown below.

```
SELECT A.agent_id, A.first, A.middle, A.last, T.team_id, T.name
FROM Agent A, TeamRel TR, Team T
WHERE A.agent_id = TR.agent_id AND TR.team_id = T.team_id
```

4. (7 points) List all possible pairs of two distinct languages. (That is, do not list a language with itself.)

$\pi_{L1.language, L2.language} (\sigma_{L1.language \neq L2.language} (\text{Language } L1 \times \text{Language } L2))$

```
SELECT L1.language, L2.language
FROM   Language L1, Language L2
WHERE  L1.language <> L2.language
```

Another way to write this query:

```
SELECT L1.language, L2.language
FROM   Language L1, Language L2
WHERE  L1.language != L2.language
```

language	language
English	Spanish
English	German
English	French
English	Chinese
English	Japanese

380 rows

Bonus: (2 points) Modify your query so that each pair of languages is only listed once. That is, if you list “English, German” then you should not list “German, English”. Show your new SQL query, (at most) five rows of your query answer, and the number of rows.

$\pi_{L1.language, L2.language} (\sigma_{L1.language \neq L2.language \text{ AND } L1.language < L2.language} (\text{Language } L1 \times \text{Language } L2))$

```
SELECT L1.language, L2.language
FROM   Language L1, Language L2
WHERE  L1.language != L2.language
      AND L1.language < L2.language
```

language	language
English	Chinese
English	Arabic
English	Cherokee
English	Bengali
Spanish	English

190 rows

5. (7 points) List mission names for missions that have an agent who speaks ‘German’ and an agent who has the skill of ‘Sniper’. (It doesn’t need to be the same agent who speaks German and who has the skill of sniper; it could be two agents – or it could be one agent. Either situation is fine; a team with either or both of these possibilities should be included in the query answer.) Show the SQL query, (at most) the first five rows of your query answer, and the number of rows in the query answer.

The important point here is that you need to join team to two different agents – so that you can ask if one is a Sniper and the other speaks German. You can also go straight to teamrel and languagerel and to skillrel; you don’t need team or agent.

$$\pi_{M.name} (\sigma_{L.language = 'German'} (((Mission M \bowtie_{team\_id=team\_id} Team T) \bowtie TeamRel TR) \bowtie Agent A) \bowtie LanguageRel LR) \bowtie Language L) \cap$$

$$\pi_{M2.name} (\sigma_{S.skill = 'Sniper'} (((Mission M2 \bowtie_{team\_id=team\_id} Team T2) \bowtie TeamRel TR2) \bowtie Agent A2) \bowtie SkillRel SR) \bowtie Skill S)$$

```
SELECT M.name
FROM Mission M JOIN Team T USING (team_id) NATURAL JOIN
      TeamRel TR NATURAL JOIN Agent A NATURAL JOIN
      LanguageRel LR NATURAL JOIN Language L
WHERE L.language = 'German'
INTERSECT
SELECT M2.name
FROM Mission M2 JOIN Team T2 USING (team_id) NATURAL JOIN
      TeamRel TR2 NATURAL JOIN Agent A2 NATURAL JOIN
      SkillRel SR NATURAL JOIN Skill S
WHERE S.skill = 'Sniper'
```

Another way to write this query:

```
SELECT M.name
FROM Mission M JOIN TeamRel TR1 ON M.team_id = TR1.team_id
      JOIN TeamRel TR2 ON M.team_id = TR2.team_id
      JOIN LanguageRel LR ON TR1.agent_id = LR.agent_id
      JOIN SkillRel SR ON SR.agent_id = TR2.agent_id
      JOIN Skill S ON S.skill_id = SR.skill_id
      JOIN Language L ON L.lang_id = LR.lang_id
WHERE L.language = 'German' AND S.skill = 'Sniper'
```

name
Black Downs
Dimrill Dale
Common Speech
Pimple
Companions

15 rows

6. (4 points) Modify the SQL query in question 5 to a skill of ‘Sniper’ and a language of ‘Greek’. Show the SQL query, (at most) the first five rows of your query answer, and the number of rows in the query answer.

```

SELECT M.name
FROM Mission M JOIN Team T USING (team_id) NATURAL JOIN
      TeamRel TR NATURAL JOIN Agent A NATURAL JOIN
      LanguageRel LR NATURAL JOIN Language L
WHERE L.language = 'Greek'
INTERSECT
SELECT M2.name
FROM Mission M2 JOIN Team T2 USING (team_id) NATURAL JOIN
      TeamRel TR2 NATURAL JOIN Agent A2 NATURAL JOIN
      SkillRel SR NATURAL JOIN Skill S

```

No rows found in this case since there is no agent who speaks Greek.

7. (4 points) Modify the SQL query in question 5 to list missions where a single agent has the skill of ‘Forgery’ and also speaks ‘Russian’. Show the SQL query, (at most) the first five rows of your query answer, and the number of rows in the query answer.

```

SELECT DISTINCT M.name
FROM Mission M JOIN Team T USING (team_id)
      NATURAL JOIN TeamRel TR
      NATURAL JOIN Agent
      NATURAL JOIN LanguageRel LR
      NATURAL JOIN Language L
      NATURAL JOIN SkillRel SR
      NATURAL JOIN Skill S
WHERE L.language = 'Russian' AND S.skill = 'Forgery'

```

name
Artano
Baldor
Battle of the Peak
Beleriand
Book of Mazarbul

50 rows

8. (14 points) List the SQL queries that you used to convince yourself that the answers you got for question 5, 6, and 7 are correct. Explain your reasoning. (You don’t need to give the entire query answer for any of these queries but you do need to show the SQL queries and you need to explain your reasoning.)

How do we know if the answer is correct?

First four rows of query answer are:

Artano
Baldor
Battle of the Peak
Beleriand

```
select m.name, sr.agent_id, l.language, s.skill
from
mission m join teamrel tr1 on m.team_id = tr1.team_id
```

```
    join skillrel sr on tr1.agent_id = sr.agent_id
    join languagerel lr on tr1.agent_id = lr.agent_id
    join language l on lr.lang_id = l.lang_id
    join skill s on s.skill_id = sr.skill_id
```

where l.language = 'Russian' and s.skill = 'Forgery' and m.name = 'Artano'

name	agent_id	language	skill
Artano	947	Russian	Forgery

Let's check out agent with id of 947. Which skills does he/she have?

Select \* from skillrel join skill using (skill\_id) where agent\_id = 947

skill_id	agent_id	skill
3	947	Driver
7	947	Chemical Expert
21	947	Computer Forensics
39	947	Forgery

4 row(s)

Let's check out agent with id of 947. Which languages does he/she speak?

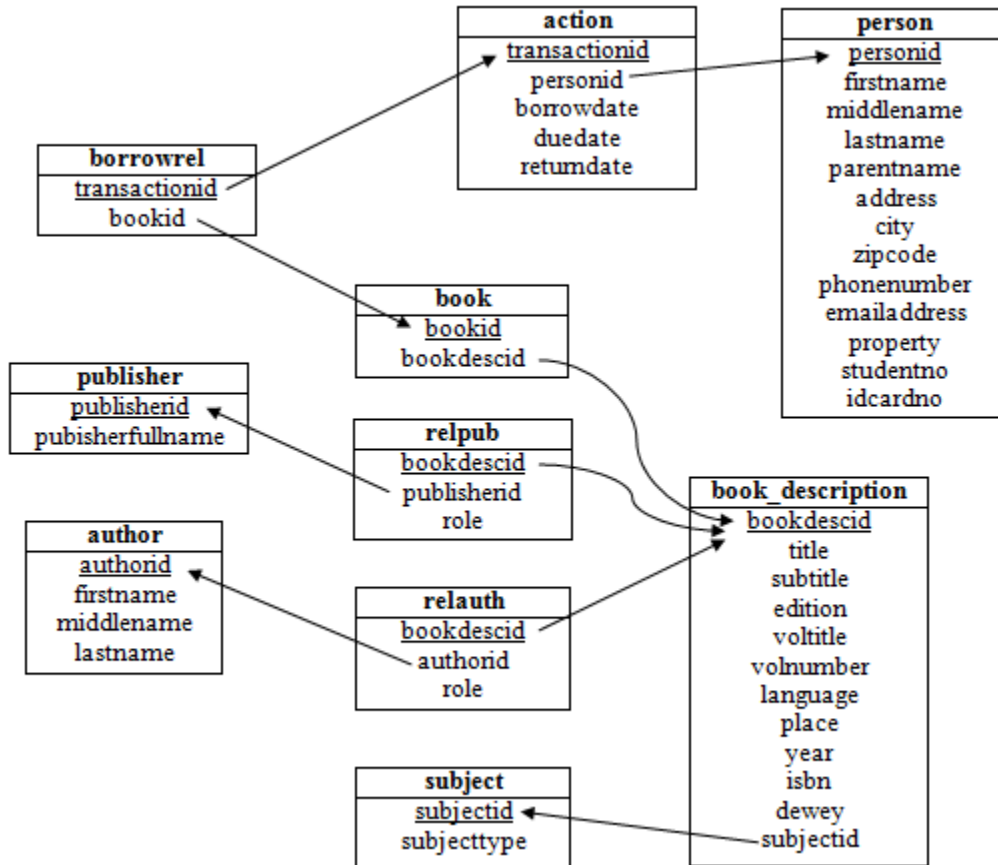
Select \* from skillrel join skill using (skill\_id) where agent\_id = 947

lang_id	agent_id	language
4	947	French
7	947	Russian
17	947	Malay
19	947	Hindi

Similarly, we can check the answer of question 5 by this way.

## Part III Outer Joins – Library DB

9. (20 points) Draw a schema diagram for the introdb\_library database – in the same style that you used for the diagram that you drew for the introdb\_spy database in assignment 1.



10. (4 points) Write an SQL query for the introdb\_library DB that lists the PublisherID for all publishers along with the BookDescriptionID for the book that they've published. Be sure to include all publishers – even if they haven't published a book in this database. Show the SQL query, (at most) the first five rows of the query answer, and the total number of rows in the query answer.

```

SELECT publisherid, bookdescid
FROM Publisher LEFT JOIN relpub USING (publisherid)

```

<b>publisherid</b>	<b>bookdescid</b>
3	520
3	518
3	495
3	473
3	451

543 rows

(3 points) Also, list the first three publishers that you see in your query answer that haven't published any books. You should identify these three publishers manually – just by looking at the query answer you get from your query.

18	NULL
19	NULL
21	NULL

11. (4 points) Write an SQL query for the `introd_b_library` DB that lists the `BookID` for all books along with the `PersonID` of the person who borrowed the book (as recorded in the `Action` table). Be sure to include all books, whether or not they have ever been borrowed. List each appropriate (`BookID`, `PersonID`) pair just once in your query answer. Show the SQL query, (at most) the first five rows of the query answer, and the total number of rows in the query answer.

```
SELECT  DISTINCT B.bookid, A.personid
FROM    Book B LEFT JOIN BorrowRel BR USING (bookid)
        LEFT JOIN Action A USING (transactionid)
ORDER BY B.bookid
```

<b>bookid</b>	<b>personid</b>
195	NULL
251	NULL
104	184
91	460
502	254

523 rows

12. (2 points) Write your SQL query from question 11 again. If you used `LEFT` join in query 11, then write the query using `RIGHT` join. (If you used `RIGHT` join in query 11, then write the query using `LEFT` join.) Show the SQL query, (at most) the first five rows of the query answer, and the total number of rows.

```

SELECT DISTINCT B.bookid, A.personid
FROM   Action A RIGHT JOIN BorrowRel BR USING (transactionid)
      RIGHT JOIN Book B USING (bookid)

```

bookid	personid
195	NULL
251	NULL
104	184
91	460
502	254

523 rows

#### Part IV GROUP BY with HAVING - SQL Queries – introdb\_spy DB

13. (4 points) List the mission\_id with the count of the number of members on the team associated with the mission, for all missions. Show the SQL query, (at most) the first five rows of your query answer, and the number of rows in the query answer.

```

SELECT  M.mission_id, COUNT (TR.agent_id)
FROM    Mission M, Team T, TeamRel TR
WHERE   M.team_id = T.team_id AND TR.team_id = T.team_id
GROUP BY M.mission_id

```

mission_id	count
129	9
195	10
251	8
106	10
120	8

404 rows

14. (4 points) List the agent\_id with the count of the number of languages that the agent speaks for all agents that speak at least five languages. Show the SQL query, (at most) the first five rows of your query answer, and the number of rows in the query answer.

```

SELECT  A.agent_id, COUNT (L.lang_id)
FROM    Agent A NATURAL JOIN LanguageRel LR NATURAL JOIN Language L
GROUP BY A.agent_id
HAVING  COUNT (L.lang_id) >= 5

```

agent_id	count
106	5
497	5
577	5
399	5
330	5

125 rows

15. (4 points) List the language id plus the count of the number of agents who speak that language, for all languages in the database. Show the SQL query and ALL of the rows in your query answer. (There should be fewer than 30 rows.)

```
SELECT LR.lang_id, COUNT (A.agent_id)
FROM Agent A NATURAL JOIN LanguageRel LR
GROUP BY LR.lang_id
```

lang_id	count
14	111
8	105
12	112
17	102
1	12
15	104
10	101
11	105
4	101
18	108
16	89
6	112
19	110
2	92
3	122
20	91
5	82
13	111
9	118
7	103

20 rows