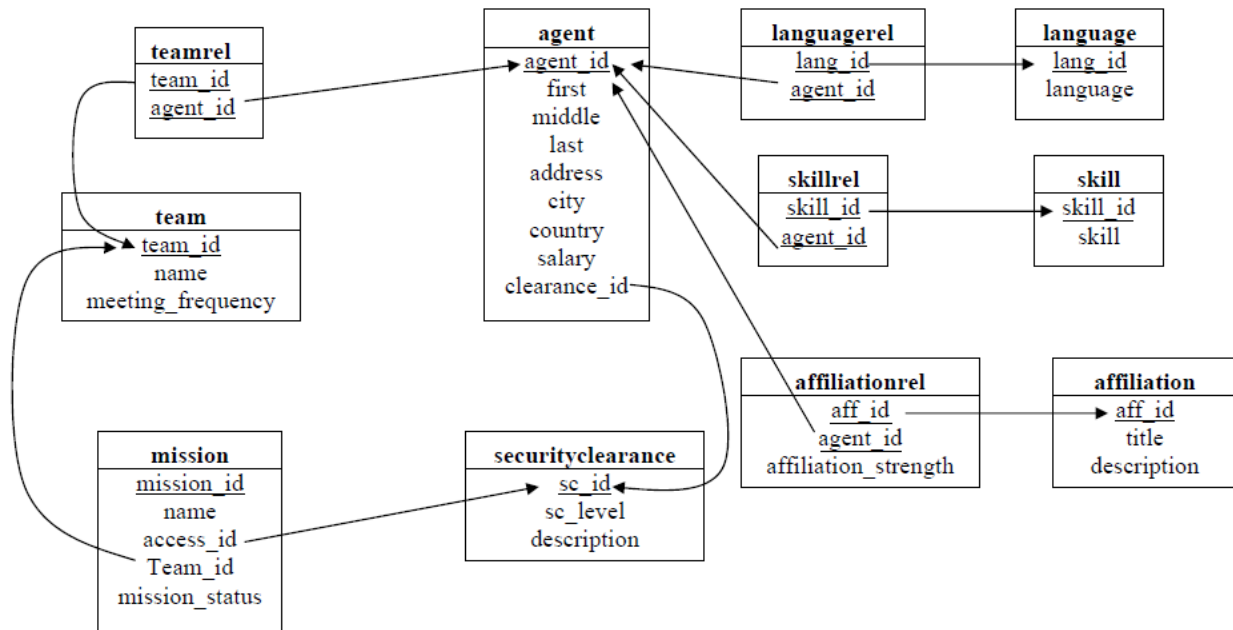


**CS 386/586 Winter 2012**  
**Assignment 1**  
**Suggested Answers**

1. (44 points) Use the introdb\_spy database available from the PostgreSQL query page as well as the description of this database available from the Database Support Page and the text file of commands that you can use to create the Spy database. These are all reachable from the class web page at <http://www.cs.pdx.edu/~lmd/cs386> Click on “link to DB information” at the top of the page; click on “PostgreSQL Query Page” or “Database Support Page”.

Draw a diagram of the schema for the Spy database using the format shown on slide 30 of Lecture 1. Each table must be represented by a box with the table name in the top row followed by the list of attributes for the table shown in the remaining rows of the box. The table key must be indicated by underlining the appropriate attributes. All foreign keys must be indicated by an arrow (in the case of a foreign key that consists of just one attribute) from the attribute that is the foreign key pointing directly to the attribute that it references. (It is NOT sufficient to simply point to the box surrounding the table; you must point directly to the attribute that is being referenced.) Hint: for the Spy database, the foreign keys always have the same name as the attribute that they reference. Hint: the foreign keys present in the tables whose names end with “Rel” are not listed in the description of the Spy database; you MUST include the foreign keys contained in these tables whose names end with Rel. Hint: if you’re not sure, look at the text file of commands that you can use to create the Spy database.

You can draw your diagram by hand or by using MS Word or MS PowerPoint or other tool.



2. (28 points) (Single table queries) Write and execute a query that finds the following answers. For each question, include the query that you executed, copy the first five rows of your query answer (or copy the entire query answer if your answer has fewer than 5 rows), and indicate the total number of rows returned.

a. List the agent\_id, first, middle, and last name, and salary for agents that have a salary < 53000.

```
SELECT agent_id, first, middle, last, salary
FROM Agent
WHERE salary < 53000
```

agent_id	first	middle	last	salary
1	Nick	Jim	Black	50553
2	Bill	NULL	Bundt	50955
7	Bill	NULL	Heeman	51564
12	Kristin	NULL	Delcambre	50503
20	George	NULL	Jones	50171

90 rows

b. List the agent\_id, first, middle, and last name, and security clearance id for agents that have a security clearance id > 5.

```
SELECT agent_id, first, middle, last, clearance_id
FROM Agent
WHERE clearance_id > 5
```

agent_id	first	middle	last	clearance_id
2	Bill	NULL	Bundt	6
4	Jim	NULL	Cowan	6
20	George	NULL	Jones	6
21	Jim	NULL	Kieburz	6
37	John	NULL	Walpole	6

38 rows

c. List the agent\_id, first, middle, and last name, city, country, and security clearance id for agents that live in Paris, France.

```
SELECT agent_id, first, middle, last, city, country, clearance_id
FROM Agent
WHERE city = 'Paris' AND country = 'France'
```

agent_id	first	middle	last	city	country	clearance_id
2	Bill	NULL	Bundt	Paris	France	6
8	Andrew	NULL	James	Paris	France	3
20	George	NULL	Jones	Paris	France	6
49	Jonathan	NULL	Hammerstrom	Paris	France	5
64	George	NULL	van Santen	Paris	France	4

54 rows

d. List the agent\_id, first, middle, and last name, security clearance id, and city for agents who live in Paris (regardless of country) and who have a security clearance id > 5.

```
SELECT agent_id, first, middle, last, clearance_id, city
FROM Agent
WHERE city = 'Paris' AND clearance_id > 5
```

agent_id	first	middle	last	clearance_id	city
2	Bill	NULL	Bundt	6	Paris
20	George	NULL	Jones	6	Paris
104	Bill	NULL	Spadaro	6	Paris
130	Nick	NULL	Fessler	6	Paris

4 rows

e. List the agent\_id, first and last names for agents who have no middle name. (Hint: to test whether an attribute has no value put this in the WHERE clause: WHERE middle is null. You can also say, for example, WHERE middle is not null – in a different query – to find agents who have a middle name.)

```
SELECT agent_id, first, last
FROM Agent
WHERE middle IS NULL
```

agent_id	first	last
2	Bill	Bundt
3	Mathew	Cohen
4	Jim	Cowan
5	George	Fairley
7	Bill	Heeman

454 rows

f. List the mission\_id, mission name, and mission status for all missions where the status is not 'success'.

```
SELECT mission_id, name, mission_status
FROM Mission
WHERE mission_status <> 'success'
```

Another way to write the query:

```
SELECT mission_id, name, mission_status
FROM Mission
WHERE mission_status != 'success'
```

mission_id	name	mission_status
2	White Crown	failed
4	Gollum	ongoing
5	Mellyrn	ongoing
6	Norland	ongoing
7	Oliphant	failed

258 rows

g. List all attributes in the team table for teams where the teams meet weekly.

```
SELECT *
FROM Team
WHERE meeting_frequency = 'weekly'
```

team_id	name	meeting_frequency
1	Renegade	weekly
2	Haberdash	weekly
7	FlyOnTheWall	weekly
9	BumbleBee	weekly
16	Vikings	weekly

12 rows

3. (28 points) (Queries that use Joins) Write and execute a query that finds the following answers. For each question, include the query that you executed, copy the first five rows of your query answer (or copy the entire query answer if your answer has fewer than 5 rows), and indicate the total number of rows returned.

a. List the agent\_id, first, middle, and last name, security clearance id, and security clearance description for agents that have a security clearance id > 5. (Note the difference between this query and query 2b above. For this one, you must join the agent table with the securityclearance table.)

```
SELECT agent_id, first, middle, last, clearance_id, description
FROM Agent A, SecurityClearance SC
WHERE A.clearance_id = SC.sc_id AND A.clearance_id > 5
```

agent_id	first	middle	last	clearance_id	description
2	Bill	NULL	Bundt	6	Second lowest clearance
4	Jim	NULL	Cowan	6	Second lowest clearance
20	George	NULL	Jones	6	Second lowest clearance
21	Jim	NULL	Kieburtz	6	Second lowest clearance
37	John	NULL	Walpole	6	Second lowest clearance

38 rows

b. List the mission id, team name, and mission status for all missions that have a status of 'failed'. (You must join the mission table with the team table.)

```
SELECT M.mission_id, T.name, M.mission_status
FROM Mission M, Team T
WHERE M.team_id = T.team_id AND M.mission_status = 'failed'
```

Two additional ways to write the query are as shown below.

```
SELECT M.mission_id, T.name, M.mission_status
FROM   Mission M JOIN Team T ON (M.team_id = T.team_id)
WHERE  M.mission_status = 'failed'
```

```
SELECT M.mission_id, T.name, M.mission_status
FROM   Mission M JOIN Team T USING (team_id)
WHERE  M.mission_status = 'failed'
```

mission_id	name	mission_status
2	SpecialForces	failed
7	Cha Cha Cha	failed
19	Terminator	failed
30	Renegade	failed
32	Boat Team 3	failed

116 rows

c. List the agent\_id, first, middle, and last name, skill id and skill name that shows exactly the skills that the agent has. (You must join the agent table to the skill\_rel table and you must join the skill\_rel table to the skill table.) Describe the queries that you issued to convince yourself that you have the right answer for this query. Describe your rationale for using those queries and how you decided that your query answer was correct.

```
SELECT A.agent_id, A.first, A.middle, A.last, S.skill_id, S.skill
FROM   Agent A, SkillRel SR, Skill S
WHERE  A.agent_id = SR.agent_id AND SR.skill_id = S.skill_id
```

Two additional ways to write the query are as shown below.

```
SELECT A.agent_id, A.first, A.middle, A.last, S.skill_id, S.skill
FROM   Agent A JOIN SkillRel SR ON (A.agent_id = SR.agent_id)
      JOIN Skill S ON (SR.skill_id = S.skill_id)
```

```
SELECT A.agent_id, A.first, A.middle, A.last, S.skill_id, S.skill
FROM   Agent A JOIN SkillRel SR USING (agent_id)
      JOIN Skill S USING (skill_id)
```

agent_id	first	middle	last	skill_id	skill
168	Richard	NULL	Venkatesh	1	Sniper
207	Kate	NULL	Rask	1	Sniper
543	Mario	NULL	Rinne	1	Sniper
22	George	NULL	Launchbury	2	Demolition Expert
36	Nick	NULL	Steere	2	Demolition Expert

1954 rows

We can write some simpler queries to convince ourselves that we got the correct answer by checking any record in the above table.

```
SELECT first, middle, last
FROM Agent
WHERE agent_id = 168
```

We can check if agent Richard Venkatesh has the agent ID 168. And the result is correct as shown below.

first	middle	last
Richard	NULL	Venkatesh

```
SELECT skill_id
FROM SkillRel SR
WHERE agent_id = 168
```

This query helps us find all the associated skill IDs with agent whose ID is 168. And the skill IDs are 1, 23, 33, 42, and 51 as shown below.

skill_id
1
23
33
42
51

Then we can check if the skill ID of skill 'Sniper' would be matched with one of the five skill IDs we just got by the following query.

```
SELECT skill_id
FROM Skill
WHERE skill = 'Sniper'
```

The result of this query is as following.

skill_id
1

This means that the agent whose ID is 168 has the skill 'Sniper'.

We can check another record in a different way. We can first find what the skill ID of skill 'Demolition Expert'.

```
SELECT skill_id
FROM Skill
WHERE skill = 'Demolition Expert'
```

The skill ID we got is 2.

skill_id
2

Then we can check if agents with IDs 22 and 36 have this skill ID.

```
SELECT agent_id
FROM SkillRel
WHERE skill_id = 2
```

And we got the result as following.

agent_id
22
36
50
101
161
175
193
204
267
278
281
312
346
350
388
517
559
703
716
723
803
821
836
1040

We can see that agent IDs 22 and 36 are in the result as expected.

And we can also check if the agent names are correct with agent IDs 22 and 36. And by checking the result with the answer of the original query, the result is correct.

```
SELECT agent_id, first, middle, last
FROM Agent
WHERE agent_id = 22 OR agent_id = 36
```

agent_id	first	middle	last
22	George	NULL	Launchbury
36	Nick	NULL	Steere

Similarly, we can convince ourselves by checking some more records in the result by this way and we can say that the original query to this question is correct.

d. List the agent\_id, first, middle, and last name, language id, and language name for all agents that have a security clearance id > 5. You must list the languages that show exactly the languages spoken by the agent.

```
SELECT A.agent_id, A.first, A.middle, A.last, L.lang_id, L.language
FROM Agent A, LanguageRel LR, Language L
WHERE A.agent_id = LR.agent_id
      AND LR.lang_id = L.lang_id
      AND A.clearance_id > 5
```

agent_id	first	middle	last	lang_id	language
37	John	NULL	Walpole	2	Spanish
86	Leonidas	NULL	Galanis	2	Spanish
704	Orrin	G	Nelson	2	Spanish
743	Lisa	J	Lautenberg	2	Spanish
872	Lucie	R	Culp	2	Spanish

110 rows

e. List the team name and first, middle, and last name for the agent. This query must show the team name for agents that are on that team.

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

Two additional ways to write the query are as shown below.

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

```
SELECT T.name, A.first, A.middle, A.last
FROM   Agent A JOIN TeamRel TR ON (A.agent_id = TR.agent_id)
       JOIN Team T ON (TR.team_id = T.team_id)
```

name	first	middle	last
Renegade	Richard	NULL	Venkatesh
Renegade	Bill	NULL	White
Renegade	George	NULL	Flood
Renegade	Nicholas	NULL	Larsen
Renegade	Crispin	NULL	Rios

374 rows

4. (5 points) Bonus question: For query 3a and for query 3d, write each query two additional ways (beyond the way you wrote the query above). The three ways should be: (a) using a JOIN clause with the ON clause in the FROM, (b) using a JOIN clause with the USING clause in the FROM, and (c) using NO join clause in the FROM clause.

For query 3a:

```
SELECT agent_id, first, middle, last, clearance_id, description
FROM   Agent A JOIN SecurityClearance SC ON
       A.clearance_id = SC.sc_id
WHERE  clearance_id > 5
```

We can't use USING clause because we need to join on two attributes that have different names.

For query 3d:

```
SELECT A.agent_id, A.first, A.middle, A.last, L.lang_id, L.language
FROM   Agent A JOIN LanguageRel LR USING (agent_id)
       JOIN Language L USING (lang_id)
WHERE  A.clearance_id > 5
```

```
SELECT A.agent_id, A.first, A.middle, A.last, L.lang_id, L.language
FROM   Agent A JOIN LanguageRel LR ON A.agent_id = LR.agent_id
       JOIN Language L ON LR.lang_id = L.lang_id
WHERE  A.clearance_id > 5
```