1. Which SQL select will display the employees' last_names and their job names?

```
a.  SELECT  employees.last_name,  jobs.job_title
     FROM     employees,  jobs
     WHERE   e.employee_id = j.job_id;
```

Cannot compare employee_id and job_id – they are two different columns

```
b. SELECT   e.last_name,   j.job_title
     FROM    employees e,   jobs  j
     WHERE  e.employee_id = j.job_id;
```

Cannot compare employee_id and job_id – they are two different columns

```
c.  SELECT  e.last_name,   j.job_title
     FROM   employees e,   jobs  j
     WHERE    e.job_id = j.job_id;
```

```
d. SELECT last_name, job_title
    FROM   employees, jobs
    WHERE  employee_id = job_id;
```

Cannot compare employee_id and job_id – they are two different columns

2. Which of the following will do the same but in SQL 92 ?

```
a. SELECT employees.last_name, jobs.jobs_title
   FROM employees INNER JOIN jobs
   USING (job_id);
```

```
b. SELECT e.last_name,   j.job_title
   FROM employees e INNER JOIN jobs j
   USING (employee_id);
```

employee_id is not common to both tables

```
c. SELECT e.last_name, j.job_title
   FROM employees e  INNER JOIN jobs j
   USING (e.employee_id = j.job_id);
```

Comparing different column attributes

```
d. SELECT last_name, job_title
   FROM employees INNER JOIN jobs
   ON (job_id);
```

The ON clause must have table1.attribute1 = table2.attribute2

3. Which SQL select will display the employee last names and employee IDs who started employment after Jan 21, 2003 in the department name of “Sales”?

```
a.  SELECT  employees.last_name, employees.employee_ID
     FROM     employees,  job_history, department
     WHERE   e.employee_id = j.employee_id
     AND      e.department_id = d.deparment_id
     AND      j.start_date > ’21-Jan-2003’
     AND      upper(d.department_name) = ‘Sales’;
```

The upper function converts text to uppercase.

```
b. SELECT   last_name,  employee_ID
     FROM    employees e, job_history  j, departments d
     WHERE   e.employee_id = j.employee_id
     AND     e.department_id = d.department_id
     AND     j.start_date > ’21-Jan-2003’
     AND     d.department_name = ‘Sales’;
```

employee_ID is defined in two tables - ambiguous

```
c.  SELECT  e.last_name, e.employee_ID
     FROM    employees e, job_history j, departments d
     WHERE   e.employee_id = j.employee_id
     AND     j.start_date > ’21-Jan-2003’
     AND     upper(d.department_name) = ‘SALES’;
```

Missing the join to the departments table

```
e. SELECT last_name, e.employee_ID
    FROM   employees e, job_history, departments
    WHERE  e.employee_id = job_history.employee_id
    AND    e.department_id = departments.department_id
    AND    start_date > ’21-Jan-2003’
    AND    department_name = ‘Sales’;
```

4. Which of the following will do the same but in SQL 92 ?

```
a. SELECT employees.last_name, employees.employee_ID
   FROM employees INNER JOIN job_history
   USING (employee_ID) INNER JOIN departments
   USING (department_ID)
   WHERE start_date > ’21-Jan-2003’
   AND   department_name = ‘Sales’;
```

department_ID is ambiguous

```
b. SELECT e.last_name, e.employee_ID
   FROM employees e INNER JOIN job_history j
   USING (employee_id) INNER JOIN departments d
   USING (job_id)
   WHERE j.start_date > ’21-Jan-2003’
   AND   d.department_name = ‘Sales’;
```

d.job_id does not exist

```
c. SELECT e.last_name, e.employee_id
   FROM employees e INNER JOIN job_history j
   ON (e.employee_id = j.employee)
   INNER JOIN departments d
   ON (d.department_id = j.employee_id)
   WHERE j.start_date > ’21-Jan-2003’
   AND   d.department_name = ‘Sales’;
```

```
d. SELECT e.last_name, employee_ID
   FROM departments d INNER JOIN employees e
   ON (d.department_id = e.department_id) INNER JOIN
   JOB_HISTORY j
   USING (employee_ID)
   WHERE j.start_date > ’21-Jan-2003’
   AND   lower(d.department_name) = ‘sales’;
```

5. Which select will NOT return the list of employees who work in Canada?

```
a. SELECT last_name AS “Last Name”
   FROM employees, departments, locations, countries
   WHERE employees.employee_id =
   departments.employee_id
   AND countries.location_id = locations.location_id
   AND departments.country_id = countries.country_id
   AND upper(country_name) = ‘CANADA’;
```

```
b. SELECT last_name “Last Name”
   FROM employees INNER JOIN departments
   USING (department_id)
   INNER JOIN locations
   USING (location_id)
   INNER JOIN countries
   USING (country_id)
   WHERE upper(country_name) = ‘CANADA’;
```

```
c. SELECT e.last_name
   FROM employees e INNER JOIN departments d
   ON (e.employee_id = d.employee_id)
   INNER JOIN locations l
   ON (d.location_id = l.location_id)
   INNER JOIN countries c
   ON (l.country_id = c.country_id)
   WHERE upper(country_name) = ‘CANADA’;
   departments table does not have employee_ID
d. SELECT last_name
   FROM employees
   WHERE department_id IN
    ( SELECT department_id
   FROM departments
   WHERE location_id IN
      ( SELECT location_id
         FROM locations
         WHERE country_id IN
          ( SELECT country_id
             FROM countries
             WHERE country_name = ‘Canada’)));
```

6. Which SELECT(s) shows the last name of current managers who at one time had been employed before (have a record in job_history) ?

```
a. select last_name
   from employees, jobs, job_history
   where employees.job_id = jobs.job_id
   and job_title like '%Manager%'
   and employees.employee_id =
job_history.employee_id;

b. select last_name
   from employees, jobs
   where employees.job_id = jobs.job_id
   and job_title like '%Manager%’
   and employee_id IN
   (select employee_id from job_history);


c. select e.last_name
   from employees e, job_history j
   where e.job_id =
   ( select job_id
     From jobs
     Where job_title like ‘%Manager%’)
   And e.employee_id = j.employee_id;
```

Subquery returns more than one row – this causes an error.

```
d. select last_name
   from employees e
```

```
where exists
     ( select job_history.job_id
         From job_history
         where e.job_id = job_history.job_id)
  and
  e.job_id in
  (select job_id
     from jobs
     Where job_title like '%Manager%')
```

This won’t select those who have a record in job_history – it should be where e.employee_id=job_history.employee_id)