3.4 (a) SELECT COUNT (Distinct person.driver_id)
FROM owns as O
WHERE O.license_plate in (SELECT license_plate
FROM accident participated WHERE accident.
report_number = participated.report_number AND
accident.year=2017)

3.11 c SELECT dept_name, max(salary) as max_salary
FROM instructor group by dept_name

d WITH dept_max(dept_name, max_salary) as
(SELECT dept_name, max(salary)) as max_salary
FROM instructor Group by dept_name)
SELECT min(max_salary) FROM dept_max

3.12 a INSERT INTO course (course_id, title,
dept_name, credits) VALUES ('CS-001', 'Weekly
Seminar', 'Comp. Sci.', 0)

b INSERT INTO section (course_id, sec_id,
semester, year) VALUES ('CS-001', '1', 'Fall', 2017)

c INSERT INTO takes (ID, course_id, sec_id, sem
SELECT ID, 'CS-001', '1', 'Fall', '2017' ster, year)
FROM student WHERE dept_name = 'Comp_Sci'

d DELETE FROM takes WHERE ID = '12345'
AND (course_id, sec_id, semester, year) = ('CS-001',
'1', 'Fall', '2017')

e DELETE FROM course WHERE course_id = 'CS-00
当删除 course 元组时，所有相关的 section 元组也
会被删除。所以删除 CS-001 时所有在 section 中
的相关行也会被删，对 course_id 引用会出错

f DELETE FROM takes WHERE course_id in
(SELECT course_id FROM course WHERE
LOWER (title) LIKE '% advanced %')

3.13 CREATE TABLE person (
driver_id int, name varchar(20) not null
address varchar(50) not null PRIMARY KEY (driver
-id))

```
CREATE TABLE car(
  license_plate char(8) model varchar(10)
  year int, PRIMARY_KEY (license_plates))

CREATE TABLE accident(
  report_number int, year int
  location varchar(50) PRIMARY_KEY (report_
  number));

CREATE TABLE owns(
  driver_id int, license_plate char(8)
PRIMARY_KEY (driver_id, license_plate) FOREIGN
_KEY (driver_id) references person FOREIGN_
KEY (license_plate) references car);

CREATE TABLE participated( report_number int,
license_plate char(8) driver_id int, damage_amount
NUMERIC (10,2), PRIMARY_KEY (report_number,
license_plate), FOREIGN_KEY (report_number) reference
accident, FOREIGN_KEY (license_plate) references car
```

FOREIGN KEY (driver_id) references person);

3.28 SELECT id, name FROM instructor as i
WHERE not exists (SELECT course_id FROM course WHERE dept_name = i.dept_name) Except
(SELECT course_id FROM teaches WHERE teaches.id = i.id) ORDER BY name ASC

4.3 a SELECT * FROM student NATURAL Join takes
UNION
SELECT ID, name, dept_name, tot_cred, null,
null, null, null, null
FROM student S1
WHERE not exists (SELECT ID FROM takes
T1 WHERE T1.id = S.id)

b (SELECT * FROM student NATURAL Join takes)
UNION
(SELECT ID, name, dept_name, tot_cred, null
null, null, null, null
FROM student S1

WHERE not exists (SELECT ID FROM takes T1
WHERE T1.id = S1.id))

UNION

(SELECT ID, null, null, null, course_id, sec_id,
semester, year, grade FROM takes T1
WHERE NOT exists (SELECT ID FROM
student S1 WHERE T1.id = S1.id)).

4.17 SELECT s.id FROM student s
LEFT OUTER JOIN advisor a ON
s.id = a.s_id WHERE a.i_id IS NULL
OR a.s_id IS NULL