1. Create a view that show only the employee id, first name, last name and email for employees earning more than 10,000 per month.

CREATE VIEW v\_EmpHighSal AS

SELECT employee\_id, first\_name, last\_name, email

FROM employees

WHERE salary > 10000;

2. Create a view that show relevant details for employees earning more than 100,000 a year.

CREATE VIEW v\_HighAnnSal

AS SELECT employee\_id, last\_name, salary\*12 ANN\_SALARY

FROM employees

WHERE (salary\*12) > 100000;

3. Create a view that will display the following when queried.

LOCATION COUNTRY REGION

-------------------------------------------------- ---------------------------------------- -------------------------

1200 Tokyo Prefecture Japan Asia

1400 Texas United States of America Americas

1500 California United States of America Americas

1600 New Jersey United States of America Americas

1700 Washington United States of America Americas

1800 Ontario Canada Americas

1900 Yukon Canada Americas

2100 Maharashtra India Asia

2200 New South Wales Australia Asia

2500 Oxford United Kingdom Europe

2600 Manchester United Kingdom Europe

where the 'LOCATION' column is location\_id and state\_province combined;

'COUNTRY' column is the name of a country and

'REGION' column is the name of a region

Another requirement is do not include records that do not have state\_province name

create or replace view v\_LocationRegion(location, country, region) as

select location\_id||' '||state\_province, country\_name, region\_name

from locations L, countries C, regions R

where (L.country\_id = C.country\_id) and

(C.region\_id = R.region\_id) and

(state\_province is not null);

For questions 4-9, substitute the equijoin in thw where clause with USING...ON...

and us the AND operator or the WHERE clause for the additional conditions.

Refer to Chapter 6 lecture notes.

10. Write a query using OUTER JOIN to produce the following result that show the previous and current number of staff in each job title:

(Hint: you need to create TWO views)

create or replace view v\_Previous\_Job as

Select J.job\_id, J.job\_title, COUNT(\*) as No\_Of\_Staff

from Jobs J, Job\_History JH

where (J.job\_id = JH.job\_id)

group by J.job\_id, J.job\_title

order by 1;

create or replace view v\_Current\_Job as

Select J.job\_id, , J.job\_title, COUNT(\*) as No\_Of\_Staff

from Jobs J, Employees E

where (J.job\_id = E.job\_id)

group by J.job\_id, J.job\_title

order by 1;

select P.No\_of\_Staff as Previous\_No,

C.No\_of\_Staff as Current\_No, C.Job\_Id, C.Job\_Title

from v\_Previous\_Job P RIGHT OUTER JOIN v\_Current\_Job C

ON (P.job\_id = C.job\_id);

\*Note: using RIGHT OUTER JOIN on the assumption that there will be more staff in the current job titles compared tp previous job titles.

Can experiment with LEFT and FULL OUTER JOIN to see the difference in results.