

1. Таблиця Employees.
Напишіть запит для вибору імен, прізвищ, окладів усіх працівників та іх id департаменту, оклад яких не перевищує 10000.

```
SELECT first_name, last_name, salary, department_id  
FROM employees  
WHERE salary <=10000
```

2. Таблиця Employees.
Напишіть запит для відображення імен, прізвищ, ідентифікатора посад та комісійних винагород всіх працівників, хо не мають права на отримання комісійної винагороди.

```
SELECT first_name, last_name, job_id, commission_pct  
FROM employees  
WHERE commission_pct IS NULL
```

3. Таблиця Employees.
Напишіть 2 варіанти запиту для відображення списку всіх імен, прізвищ та іх id департаменту, де їм є співробітника John.

```
SELECT first_name, last_name, department_id  
FROM employees  
WHERE first_name='John'
```

```
SELECT first_name, last_name, department_id  
FROM employees  
WHERE first_name LIKE 'John'
```

4. Таблиця Employees.
Напишіть запит для відображення імен, прізвищ співробітників із ЗП із 80-го департаменту і ЗП менше 10000.

```
SELECT first_name, last_name, salary, department_id  
FROM employees  
WHERE department_id=80 AND salary<10000
```

5. Таблиця Employees.
Напишіть запит для відображення імен, прізвищ співробітників із ЗП із 80-го і із 110 департаменту, ЗП яких знаходиться в діапазоні від 10000 до 17000.

```
SELECT first_name, last_name, salary, department_id  
FROM employees  
WHERE department_id IN (80, 110)  
AND salary BETWEEN 10000 AND 17000
```

6. Таблиця Employees.
Напишіть запит для відображення імен, прізвищ співробітників із ЗП із 80-го і із 110 департаменту, ЗП яких знаходиться в діапазоні від 10000 до 17000 і які отримують комісію.

```
SELECT first_name, last_name, salary, department_id  
FROM employees  
WHERE department_id IN (80, 110)  
AND salary BETWEEN 10000 AND 17000  
AND commission_pct IS NOT NULL;
```

7. Таблиця Employees.
Напишіть запит для відображення списку всіх імен, прізвищ співробітників, у котої ім'я починається на D та підпорядковується менеджеру 108.

```
SELECT first_name, last_name
FROM employees
WHERE first_name LIKE 'D%' AND manager_id=108;
```

8. Таблиця Employees.
Напишіть запит для відображення всіх даних співробітників, хто підпорядковується менеджерам 100, 108, 124.

```
SELECT * FROM employees
WHERE manager_id IN (100, 108, 124);
```

9. Таблиця Employees.
Напишіть запит для відображення всіх даних співробітників, хто не отримує комісію та має ЗП менше 10000 та відсортуйте вибірку за зменшеним ЗП. Прізвище і ім'я відобразіть в одному полі, яке називати "List of employees for review".
Напишіть 2 варіанти запиту використовуючи різні оператори обєднання поля.

```
SELECT first_name || ' ' || last_name AS List_of_employees_for_review, salary
FROM employees
WHERE commission_pct IS NULL AND salary<10000
ORDER BY salary DESC;
```

SELECT CONCAT(first_name, CONCAT(' ', last_name)) AS List_of_employees_for_review, salary
FROM employees
WHERE commission_pct IS NULL AND salary<10000
ORDER BY salary DESC;

1. Таблиця DEPARTMENTS.

Застосовуючи оператор CASE напишіть скрипт, який буде виводити список назв департаментів та нове поле "State", яке буде заповнюватися в залежності від того, яке значення має поле LOCATION_ID.

Для LOCATION_ID 1700 нове поле "State" має містити запис

"Washington",

Для 1400 - "Texas"

Для 1500 і 2500 - "California"

Для 1800 - "Ontario"

2400 - "Other"

2700 - "Bavaria"

```
SELECT department_name,
CASE
WHEN location_id LIKE 1700 THEN 'Washington'
WHEN location_id LIKE 1400 THEN 'Texas'
WHEN location_id IN (1500,2500) THEN 'California'
WHEN location_id LIKE 1800 THEN 'Ontario'
WHEN location_id LIKE 2400 THEN 'Other'
WHEN location_id LIKE 2700 THEN 'Bavaria'
ELSE null
END AS state
FROM departments;
```

```
SELECT department_name,
CASE
WHEN location_id = 1700 THEN 'Washington'
WHEN location_id = 1400 THEN 'Texas'
WHEN location_id IN (1500, 2500) THEN 'California'
WHEN location_id = 1800 THEN 'Ontario'
WHEN location_id = 2700 THEN 'Bavaria'
WHEN location_id = 2400 THEN 'Other'
ELSE null
END AS state
FROM DEPARTMENTS;
```

The screenshot shows the Oracle SQL Developer interface. In the Connections pane, there is a connection named 'HR' which includes tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, etc. The Worksheet pane contains a SQL query:

```
SELECT department_name,
CASE
WHEN location_id LIKE 1700 THEN 'Washington'
WHEN location_id LIKE 1400 THEN 'Texas'
WHEN location_id IN (1500,2500) THEN 'California'
WHEN location_id LIKE 1800 THEN 'Ontario'
WHEN location_id LIKE 2400 THEN 'Other'
WHEN location_id LIKE 2700 THEN 'Bavaria'
ELSE null
END AS state
FROM departments;
```

The Query Result pane displays the output of the query:

DEPARTMENT_NAME	STATE
1 Administration	Washington
2 Marketing	Ontario
3 Purchasing	Washington
4 Human Resources	Other
5 Shipping	California
6 IT	Texas
7 Public Relations	Bavaria

2. Таблиця LOCATIONS.

Напишіть запит для відображення інформації про кількість локацій компанії у кожному COUNTRY_ID та відсортуйте список від країни із найбільшою кількістю локацій компанії.

```
SELECT COUNT (location_id), country_id
FROM locations
GROUP BY country_id
ORDER BY COUNT (location_id) DESC;
```

```
SELECT COUNT (location_id)
FROM LOCATIONS
GROUP BY COUNTRY_ID
ORDER BY COUNT (location_id) DESC;
```

The screenshot shows the Oracle SQL Developer interface. In the Connections pane, there is a connection named 'HR' which includes tables like LOCATIONS, DEPARTMENTS, etc. The Worksheet pane contains a SQL query:

```
SELECT COUNT (location_id), country_id
FROM locations
GROUP BY country_id
ORDER BY COUNT (location_id) DESC;
```

The Query Result pane displays the output of the query:

COUNT(location_id)	COUNTRY_ID
1	4 US

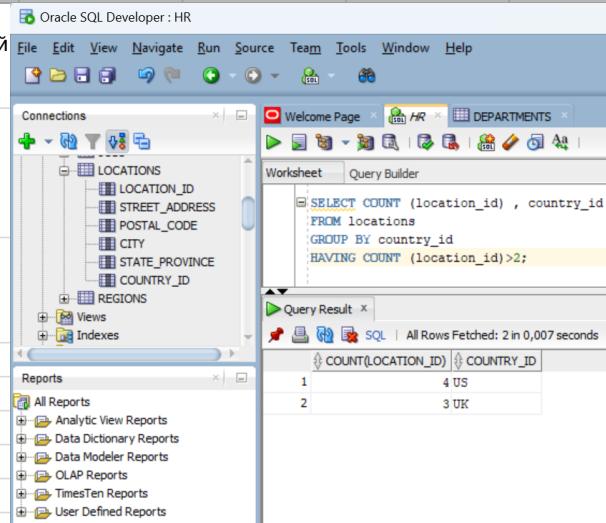
2	3 UK			
3	2 CA			
4	2 JP			
5	2 CH			
6	2 IT			
7	1 BR			
8	1 AU			
9	1 IN			
10	1 DE			
11	1 SG			
12	1 NL			

3. Таблиця LOCATIONS.

Напишіть запит для відображення інформації про кількість локацій компанії у кожному COUNTRY_ID, вивівши інформацію тільки по країнам, де кількість локацій більше двох

```
SELECT COUNT (location_id) , country_id  
FROM locations  
GROUP BY country_id  
HAVING COUNT ( location_id ) > 2;
```

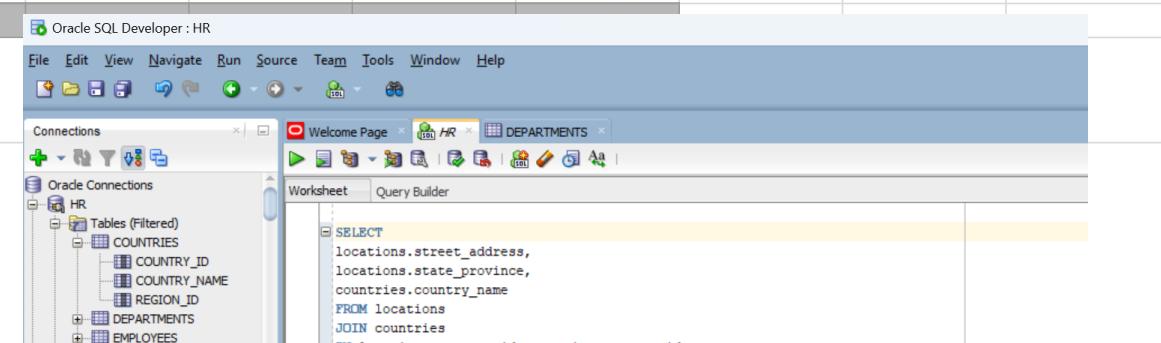
```
SELECT COUNT (location_id)  
FROM LOCATIONS  
GROUP BY COUNTRY_ID  
HAVING COUNT (location_id) >2
```



4. Таблиці LOCATIONS та COUNTRIES.

Напишіть запит для відображення всіх значень полів STREET_ADDRESS, STATE_PROVINCE та COUNTRY_NAME та відсортуйте за алфавітом (від A до Z) назву країн.

```
SELECT
locations.street_address,
locations.state_province,
countries.country_name
FROM locations
JOIN countries
ON locations.country_id=countries.country_id
ORDER BY country_name ASC;
```



```

SELECT locations.street_address, locations.state_province, countries.country_name
FROM locations
FULL JOIN countries
ON locations.country_id = countries.country_id
ORDER BY country_name ASC;

```

The screenshot shows the Oracle SQL Developer interface with a query window and a results window.

Query Window:

```

ON locations.country_id=countries.country_id
ORDER BY country_name ASC;

```

Results Window:

STREET_ADDRESS	STATE_PROVINCE	COUNTRY_NAME
1 12-98 Victoria Street	New South Wales	Australia
2 Rua Frei Caneca 1360	Sao Paulo	Brazil
3 147 Spadine Ave	Ontario	Canada
4 6092 Boxwood St	Yukon	Canada
5 40-5-12 Laogianggen	(null)	China
6 Schwanthalerstr. 7031	Bavaria	Germany
7 1298 Vileparle (E)	Maharashtra	India
8 1297 Via Cola di Rie	(null)	Italy

5. Таблиці LOCATIONS та COUNTRIES.

Напишіть запит для розрахунку кількості STREET_ADDRESS за кожною із COUNTRY_NAME, які мають відповідні значення в обох таблицях. Виведіть усі показники.

```

SELECT countries.country_name,
COUNT(locations.street_address)
FROM locations
JOIN countries
ON locations.country_id=countries.country_id
GROUP BY country_name

```

```

SELECT COUNT(locations.street_address), countries.country_name
FROM locations
JOIN countries
ON locations.country_id = countries.country_id
GROUP BY countries.country_name;

```

The screenshot shows the Oracle SQL Developer interface with a query window and a results window.

Query Window:

```

SELECT countries.country_name, COUNT(locations.street_address)
FROM locations
JOIN countries
ON locations.country_id=countries.country_id
GROUP BY country_name;

```

Results Window:

COUNTRY_NAME	COUNT(locations.street_address)
1 Brazil	1
2 Italy	2
3 Japan	2
4 Singapore	1
5 Netherlands	1
6 Germany	1
7 Mexico	1
8 United States of America	4
9 Australia	1
10 Canada	2
11 China	1

6. Таблиці LOCATIONS, COUNTRIES, REGIONS.

Напишіть запит для відображення всіх значень полів, STATE_PROVINCE та COUNTRY_NAME, REGION_NAME.

```
SELECT
locations.state_province,
countries.country_name,
regions.region_name
FROM locations
FULL JOIN countries
ON locations.country_id=countries.country_id
FULL JOIN regions
ON countries.region_id=regions.region_id ;
```

```
SELECT locations.state_province, countries.country_name, regions.region_name
FROM locations
FULL JOIN countries
ON locations.country_id = countries.country_id
FULL JOIN regions
ON countries.region_id = regions.region_id;
```

STATE_PROVINCE	COUNTRY_NAME	REGION_NAME
1 (null)	Argentina	Americas
2 New South Wales	Australia	Asia
3 (null)	Belgium	Europe
4 Sao Paulo	Brazil	Americas
5 Ontario	Canada	Americas
6 Yukon	Canada	Americas
7 Geneve	Switzerland	Europe
8 BE	Switzerland	Europe
9 (null)	China	Asia
10 Bavaria	Germany	Europe
11 (null)	Danmark	Europe

7. Таблиці LOCATIONS, COUNTRIES, REGIONS.

Напишіть запит для розрахунку кількості STREET_ADDRESS кількості COUNTRY_NAME для кожного REGION_NAME, які мають відповідні значення в усіх трьох таблицях таблицях.
Виведіть усі показники.

```
SELECT COUNT ( locations.street_address ),
COUNT (countries.country_name) , regions.region_name
FROM locations
JOIN countries ON locations.country_id=countries.country_id
JOIN regions ON countries.region_id=regions.region_id
GROUP BY regions.region_name;
```

```
SELECT COUNT (locations.street_address), COUNT (countries.country_name), regions.region_name
FROM locations
JOIN countries
ON locations.country_id = countries.country_id
JOIN regions
ON countries.region_id = regions.region_id
GROUP BY regions.region_name;
```

COUNT(locations.street_address)	COUNT(countries.country_name)	REGION_NAME
9	9	Europe
8	8	Americas
6	6	Asia

8. Таблиці LOCATIONS, COUNTRIES, REGIONS.

Напишіть запит для розрахунку кількості STREET_ADDRESS, кількості COUNTRY_NAME для кожного REGION_NAME, які мають відповідні значення в усіх трьох таблицях. Виведіть усі показники. Напишіть скрипт використовуючи аliasи для назв таблиць.

```
SELECT COUNT (l.street_address), COUNT (c.country_name),
r.region_name
FROM locations l
JOIN countries c
ON l.country_id=c.country_id
JOIN regions r
ON c.region_id=r.region_id
GROUP BY r.region_name;
```

```
SELECT COUNT (a.street_address), COUNT (b.country_name), c.region_name
FROM locations a
JOIN countries b
ON a.country_id = b.country_id
JOIN regions c
ON b.region_id = c.region_id
GROUP BY c.region_name;
```

COUNT(L.STREET_ADDRESS)	COUNT(C.COUNTRY_NAME)	REGION_NAME
1	9	Europe
2	8	Americas
3	6	Asia

9. Таблиці LOCATIONS, COUNTRIES, REGIONS.

Напишіть запит для розрахунку кількості STREET_ADDRESS кількості COUNTRY_NAME для кожного REGION_NAME, які мають відповідні значення в усіх трьох таблицях. Та виведіть ті REGION_NAME, де кількість COUNTRY_NAME буде більше 6.

```
SELECT COUNT (locations.street_address),
COUNT (countries.country_name),regions.region_name
FROM locations
JOIN countries ON locations.country_id=countries.country_id
JOIN regions ON countries.region_id=regions.region_id
GROUP BY region_name
HAVING COUNT (countries.country_name)> 6;
```

```
SELECT COUNT (a.street_address), COUNT (b.country_name), c.region_name
FROM locations a
JOIN countries b
ON a.country_id = b.country_id
JOIN regions c
ON b.region_id = c.region_id
GROUP BY c.region_name
HAVING COUNT (b.country_name)>6;
```

COUNT(LOCATIONS.STREET_ADDRESS)	COUNT(COUNTRIES.COUNTRY_NAME)	REGION_NAME
1	9	Europe
2	8	Americas

10. Таблиці LOCATIONS, COUNTRIES, REGIONS.

Напишіть запит для розрахунку кількості STREET_ADDRESS кількості COUNTRY_NAME для кожного REGION_NAME, які мають відповідні значення в усіх трьох таблицях. Та виведіть ті REGION_NAME, де кількість COUNTRY_NAME буде більше 6. Виведіть усі показники. Напишіть скрипт використовуючи аliasи для назв полів

```
SELECT
COUNT(locations.street_address) AS amount_of_street_addresses,
COUNT(countries.country_name) AS amount_of_country_names,
regions.region_name AS region
FROM locations
JOIN countries
ON locations.country_id=countries.country_id
JOIN regions
ON countries.region_id=regions.region_id
GROUP BY region_name
HAVING COUNT(countries.country_name)> 6 ;
```

```
SELECT COUNT(a.street_address) AS street, COUNT(b.
country_name) AS country, c.region_name AS region
FROM locations a
JOIN countries b
ON a.country_id = b.country_id
JOIN regions c
ON b.region_id = c.region_id
GROUP BY c.region_name
HAVING COUNT(b.country_name)>6;
```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists 'DEPARTMENTS' and 'HR'. The 'Worksheet' tab contains the SQL query:

```
SELECT COUNT(locations.street_address) AS amount_of_street_addresses,
       COUNT(countries.country_name) AS amount_of_country_names,
       regions.region_name AS region
  FROM locations
  JOIN countries
    ON locations.country_id=countries.country_id
  JOIN regions
    ON countries.region_id=regions.region_id
 GROUP BY region_name
 HAVING COUNT(countries.country_name)> 6;
```

The 'Script Output' tab shows the results:

AMOUNT_OF_STREET_ADDRESSES	AMOUNT_OF_COUNTRY_NAMES	REGION
1	9	Europe
2	8	Americas

1. Таблиці JOB_HISTORY, EMPLOYEES

Оператор UNION

Напишіть запит в якому будуть відображатися поля ідентифікатор співробітника, дата найму та дата звільнення. Якщо співробітник досі працює в компанії дата звільнення має бути null.

EMPLOYEE_ID	HIRE_DATE	END_DATE	JOB_ID
1	10017.06.2003 (null)		AD_PRES
2	10121.09.1997 27.10.2001		AC_ACCOUNT
3	10128.10.2001 15.03.2005		AC_MGR
4	10121.09.2005 (null)		AD_VP
5	10213.01.2001 24.07.2006		IT_PROG
6	10213.01.2001 (null)		AD_VP
7	10303.01.2000 (null)		IT_PROG
8	10421.05.2007 (null)		IT_PROG
9	10525.06.2005 (null)		IT_PROG
10	10605.02.2006 (null)		IT_PROG
11	10707.02.2007 (null)		IT_PROG
12	10817.08.2002 (null)		FI_MGR

1. SELECT employee_id, hire_date, null as end_date, job_id

FROM employees
UNION

SELECT employee_id, start_date, null as end_date, job_id
FROM job_history;

2. Таблиці JOB_HISTORY, EMPLOYEES

Оператор UNION

Напишіть запит в якому буде відображатися статус співробітника (status) із варіантами - заповнення current_employees - якщо співробітник досі працює в компанії і fired_employees - якщо співробітник звільнився та кількість співробітників за кожним із цих статусів

2. SELECT 'current_employees' AS status, COUNT (employee_id)
FROM employees
UNION
SELECT 'fired_employees' AS status, COUNT (employee_id)
FROM job_history;

SELECT 'current_employees' AS employee_status, COUNT(*) AS total
FROM employees e
WHERE e.employee_id NOT IN (SELECT employee_id FROM job_history)
UNION
SELECT 'fired_employees' AS employee_status, COUNT(*) AS total
FROM job_history

STATUS	COUNT(EMPLOYEE_ID)
1 current_employees	107
2 fired_employees	10

3. Таблиці JOB_HISTORY, EMPLOYEES

Оператор UNION

Напишіть запит в якому буде відображатися статус співробітника (status) із варіантами - заповнення current_employees - якщо співробітник досі працює в компанії і fired_employees - якщо співробітник звільнився та кількість співробітників за кожним із цих статусів групуючи ці показники за ідентифікатором посади.

EMPLOYEE_ID	COUNT_OF_EMPLOYEES	STATUS
1	1	current_employees
2	1	current_employees
3	1	current_employees
4	1	current_employees
5	1	current_employees
6	1	current_employees
7	1	current_employees
8	1	current_employees
9	1	current_employees
10	1	current_employees
11	1	current_employees
12	1	current_employees

```

SELECT 'current_employees' AS employee_status, job_id, COUNT(*) AS total
FROM EMPLOYEES
WHERE employee_id NOT IN (SELECT employee_id FROM JOB_HISTORY)
GROUP BY job_id
UNION
SELECT 'fired_employees' AS employee_status, job_id, COUNT(*) AS total
FROM JOB_HISTORY
GROUP BY job_id;

```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Worksheet - Query Builder:** Displays the query from the previous code block.
- Script Output x | Query Result x:** Shows the results of the query, which includes 12 rows of data mapping employee status to job IDs and their counts.

EMPLOYEE_STATUS	JOB_ID	TOTAL
1 current_employees	AC_ACCOUNT	1
2 current_employees	AC_MGR	1
3 current_employees	AD_PRES	1
4 current_employees	FI_ACCOUNT	5
5 current_employees	FI_MGR	1
6 current_employees	HR_REF	1
7 current_employees	IT_PROG	5
8 current_employees	MC_REF	1
9 current_employees	PR_REF	1
10 current_employees	PU_CLERK	5
11 current_employees	SA_MAN	5
12 current_employees	SA_REF	29
13 current_employees	ST_CLERK	50

4. Таблиці JOB_HISTORY, EMPLOYEES.

SUBQUERY

Напишть запит в якому буде відображатися за зменшеннем кількість співробітників за кожним департаментом, лише серед тих департаментів, де було звільнено більше 1 співробітника.

```

4. SELECT COUNT(employee_id), department_id
   FROM employees
  WHERE department_id IN (SELECT department_id
                           FROM job_history
                          GROUP BY department_id
                         HAVING COUNT(employee_id) > 1)
  GROUP BY department_id
 ORDER BY count(employee_id) desc;

```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Worksheet - Query Builder:** Displays the query from the previous code block.
- Script Output x | Query Result x:** Shows the results of the query, which includes 4 rows of data mapping department IDs to the count of employees fired.

COUNT(EMPLOYEE_ID)	DEPARTMENT_ID
1	45
2	34
3	3
4	2

5. Таблиці JOB_HISTORY, EMPLOYEES.

SUBQUERY

Напишть запит в якому буде відображатися ідентифікатори співробітників, імена, прізвища тільки із тих департаментів, де є співробітники, які працювали менше року

```

5. SELECT employee_id, first_name, last_name
   FROM employees
  WHERE department_id IN (SELECT department_id
                           FROM job_history
                          WHERE end_date - start_date < 365);

```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Worksheet - Query Builder:** Displays the query from the previous code block.
- Script Output x | Query Result x:** Shows the results of the query, which includes 9 rows of data mapping employee IDs to their first and last names.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
1	120 Matthew	Weiss
2	121 Adam	Tripp
3	122 Fayam	Kaufling
4	123 Shanta	Vollman
5	124 Kevin	Mourgos
6	125 Julia	Nayer
7	126 Irene	Mikkilineni
8	127 James	Landry
9	128 Stevannie	Mankin

6. Таблиці JOB_HISTORY, EMPLOYEES.

SUBQUERY

Напишіть запит в якому буде відображатися ідентифікатори співробітників, імена, прізвища тільки за умови, ящо в цьому департаменті є співробітники, які пропрацювали більше року

```
6. SELECT employee_id, first_name, last_name, department_id
   FROM employees
 WHERE EXISTS (SELECT department_id
                  FROM job_history
                 WHERE end_date - start_date > 365);

SELECT e.employee_id, e.first_name, e.last_name
   FROM hr.employees e
 WHERE e.department_id IN (
    SELECT jh.department_id
      FROM hr.job_history jh
     WHERE jh.end_date - jh.start_date > 365
   );

SELECT employee_id, first_name, last_name, department_id
   FROM employees
 WHERE department_id IN (SELECT DISTINCT department_id
                           FROM job_history
                          WHERE end_date - start_date > 365);

```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
1	201 Michael	Hartstein
2	202 Pet	Fay
3	120 Matthew	Weiss
4	121 Adam	Tripp
5	122 Payam	Kaufling
6	171 Shanta	VonLilien

7. Таблиці JOB_HISTORY, EMPLOYEES.

SUBQUERY

Напишіть запит в якому буде відображатися кількість співробітників за кожним із департаментів тільки за умови, ящо в цьому департаменті є співробітники, які пропрацювали більше року. Відсортуйте всі записи за зменшенням кількості співробітників.

```
7. SELECT COUNT(employee_id), department_id
      FROM employees
     WHERE EXISTS (SELECT department_id
                   FROM job_history
                  WHERE end_date - start_date > 365)
   GROUP BY department_id
 ORDER BY COUNT(employee_id) DESC;

SELECT COUNT(employee_id) AS count, department_id
   FROM employees
  WHERE department_id IN (SELECT DISTINCT department_id
                           FROM job_history
                          WHERE end_date - start_date > 365)
 GROUP BY department_id
 ORDER BY count DESC;

SELECT COUNT (employee_id), department_id
   FROM employees
  GROUP BY department_id
 HAVING department_id IN (SELECT department_id
                           FROM job_history
                          WHERE (END_DATE - START_DATE) > 365)
 ORDER BY COUNT (employee_id) DESC;
```

DEPARTMENT_ID	EMPLOYEE_COUNT
1	50
2	60
3	90
4	110
5	20

8. Таблиця EMPLOYEES

Напишіть скрипт, який має додавати запис у таблицю із вашим іменем і прізвищем. Дата наяву завжди має бути сьогоднішнім числом, зп = 25000, департамент = 110, менеджер 205. Інші поля можуть бути заповнені довільно.

```
8. INSERT INTO EMPLOYEES
VALUES (777, 'Iryna', 'Milinevych', 'irynamil@example.co', '0123456789',
        14.06.2024, 'IT_PROG', 25000, NULL, 205, 110);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	777 Iryna	Milinevych	0123456789	iryna...	14.06.2024	IT_PROG	25000	(null)	205	110

9. Таблиця EMPLOYEES.

Напишіть скрипт, який має оновлювати створений вище запис у таблиці із вашим іменем і прізвищем. Змініть у цьому записі департамент на 90, а ЗП на 30000.

```
UPDATE employees
SET DEPARTMENT_ID = 90, salary = 30000
WHERE employee_id = 777;
```

10. Таблиця EMPLOYEES

Напишіть скрипт, який має видаляти створений вами запис у таблиці.

```
DELETE
FROM employees
WHERE employee_id = 777;
```

```
DELETE
FROM JOB_HISTORY
WHERE EMPLOYEE_ID = 777;
```

Oracle SQL Developer : Table HR.EMPLOYEES@HR

File Edit View Navigate Run Team Tools Window Help

Connections Oracle Connections HR Tables (Filtered) COUNTRIES DEPARTMENT EMPLOYEES

EMPLOYEES HR~I Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	Iryna	Milinovich	0123456789	iryna...@example.co	14.06.2024	IT_PROG	30000	(null)	205	90

Oracle SQL Developer : HR-1

File Edit View Navigate Run Source Team Window Help

Connections Oracle Connections HR Tables (Filtered) COUNTRIES DEPARTMENT EMPLOYEES EMPLOYEE FIRST_NAME LAST_NAME EMAIL PHONE_NUMBER HIRE_DATE

Worksheet Query Builder

```
INSERT INTO EMPLOYEES
VALUES (777, 'Iryna', 'Milinovich', 'irynamil@example.co', '0123456789',
'14.06.2024', 'IT_PROG', 25000, NULL, 205, 110);

UPDATE employees
SET DEPARTMENT_ID = 90, salary = 30000
WHERE employee_id = 777;

DELETE
FROM employees
WHERE employee_id = 777;

DELETE FROM JOB_HISTORY
WHERE EMPLOYEE_ID = 777;
```

Query Result Script Output

```
DELETE
FROM employees
WHERE employee_id = 777
Error report -
ORA-02392: integrity constraint (HR.JHIST_EMP_FK) violated - child record found

1 row deleted.
```