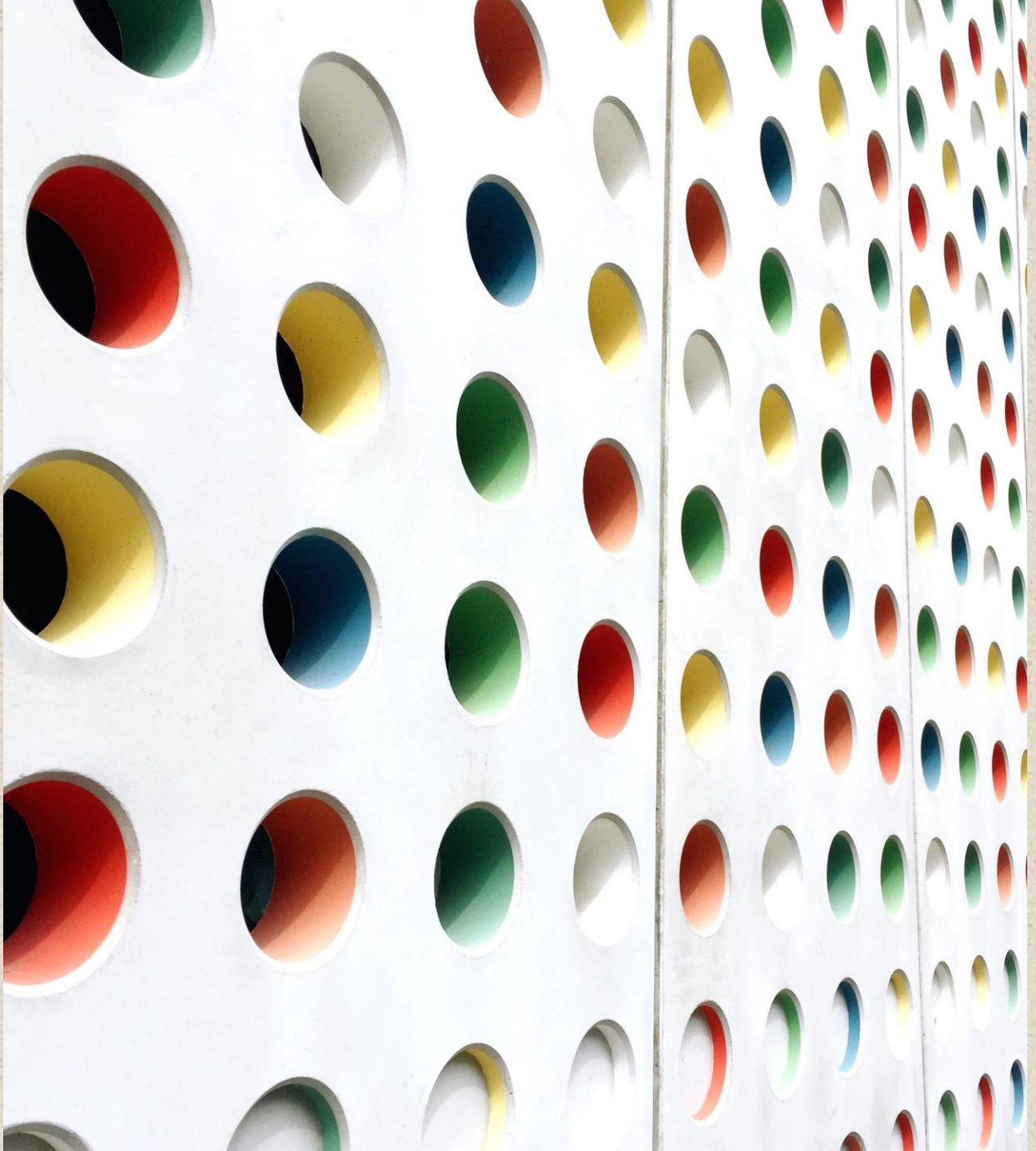


Oracle Database 12c SQL

SOLVING BUSINESS PROBLEMS

Ina Olsanskaja

Business Questions

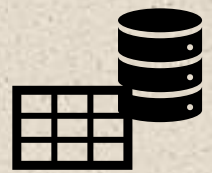
1. **LAY OFF THE PART OF THE EMPLOYEES** for savings to survive this hard COVID-19 time

2. **RELOCATE THE PART OF THE EMPLOYEES** from departments with the highest number of employees

Source: https://www.cdc.gov/coronavirus/2019-ncov/downloads/community/shelters-protect-common-areas.pdf

Implementation Steps

1. Tables and relations between them
2. Employees in each department
   Departments locations
3. Missing values
4. Finding out missing data
5. Departments with high total salary amount
6. Salary distribution in those departments
7. Employees to be laid off
8. Savings per month and year
9. List of emails to inform employees
10. Employees relocation



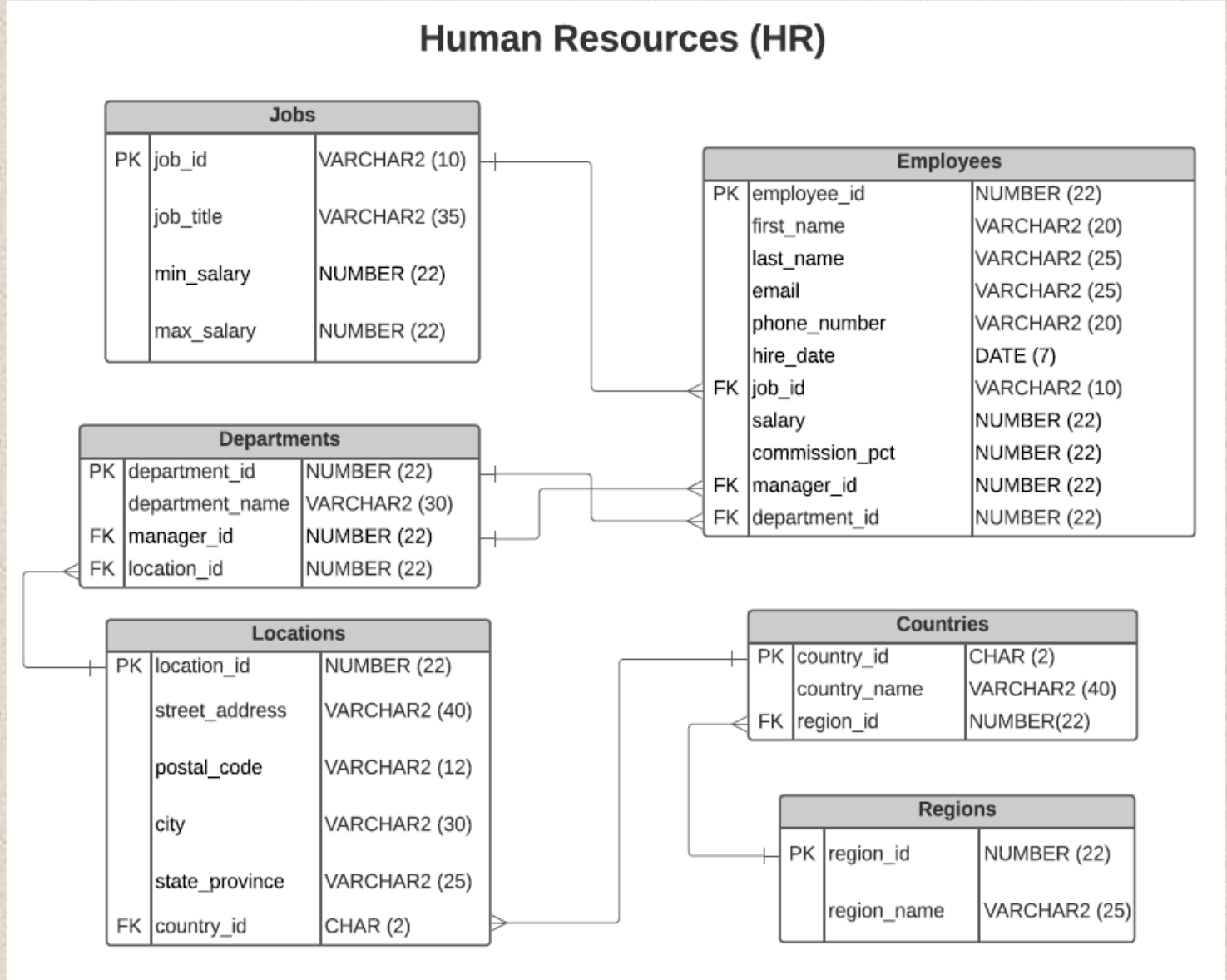






1. Schema

- **This schema helps to understand:**
  - **relations between tables**
  - **information gathered in the tables**
  - **data types**



2. Employees, Departments, Distributing by Departments

- We have 107 employees and 27 departments
- **One employee does not have department_id**
- 16 departments do not have any employees
- All 107 employees work in 11 departments
- Departments are in 7 different locations, but **each department is only in one location**

```sql
-- 1. HOW MANY EMPLOYEES DO WE HAVE?
select count(*) as EMPLOYEES_TOTAL
from hr.employees
;


-- 2. DEPARTMENTS LIST
select department_id
    , department_name
    , location_id
from hr.departments
order by location_id
;


-- 3. EMPLOYEES NUMBER IN DEPARTMENT
select count(distinct employee_id) as EMPLOYEES_TOTAL
    , department_id
from hr.employees
group by department_id
order by department_id
;


-- 4. JOINING DEPARTMENTS TO EMPLOYEES AND LOOKING FOR EMPTY DEPARTMENTS
select hr.departments.department_id
    , department_name
    , employee_id
from hr.departments
left join hr.employees
    on hr.departments.department_id=hr.employees.department_id
where employee_id is null
;

```

1.

EMPLOYEES_TOTAL
107

2.

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
60	IT	1400
50	Shipping	1500
30	Purchasing	1700
10	Administration	1700
140	Control And Credit	1700
260	Recruiting	1700
250	Retail Sales	1700
240	Government Sales	1700
90	Executive	1700
100	Finance	1700
110	Accounting	1700
120	Treasury	1700
130	Corporate Tax	1700
270	Payroll	1700
150	Shareholder Services	1700
160	Benefits	1700
170	Manufacturing	1700
180	Construction	1700
190	Contracting	1700
200	Operations	1700
210	IT Support	1700
220	NOC	1700
230	IT Helpdesk	1700
20	Marketing	1800
40	Human Resources	2400
80	Sales	2500
70	Public Relations	2700

Download CSV
27 rows selected.

3.

EMPLOYEES_TOTAL	DEPARTMENT_ID
1	10
2	20
6	30
1	40
45	50
5	60
1	70
34	80
3	90
6	100
2	110
1	-

Download CSV
12 rows selected.

4.

DEPARTMENT_ID	DEPARTMENT_NAME	EMPLOYEE_ID
120	Treasury	-
130	Corporate Tax	-
140	Control And Credit	-
150	Shareholder Services	-
160	Benefits	-
170	Manufacturing	-
180	Construction	-
190	Contracting	-
200	Operations	-
210	IT Support	-
220	NOC	-
230	IT Helpdesk	-
240	Government Sales	-
250	Retail Sales	-
260	Recruiting	-
270	Payroll	-

Download CSV
16 rows selected.

3. Dealing With Missing Values

- We have missing values in Steven King's and Kimberly Grant's records
- Manager_id we do not use in our computations
- **S.King does not have any manager_id. He is the president of the company**
- K.Grant has a lot of missing data, but we know that he is a Sales Representative, and his salary is 7000
- **Salary is important for us**
- **Let's see if we can find out all needed information about K.Grant**

```
35
36  -- 5. MISSING VALUES IN OUR TABLES
37
38  -- We will see all employees and all departments and
39  -- all missing ID values in one place
40
41  select distinct employee_id
42          , last_name
43          , first_name
44          , hr.employees.department_id
45          , department_name
46          , hr.employees.job_id
47          , job_title
48          , hr.employees.manager_id
49          , hr.locations.location_id
50          , hr.countries.country_id
51          , country_name
52          , hr.regions.region_id
53          , region_name
54          , hire_date
55  from hr.employees
56
57  left join hr.departments
58      on hr.employees.department_id=hr.departments.department_id
59  left join hr.jobs on hr.employees.job_id=hr.jobs.job_id
60  left join hr.locations
61      on hr.departments.location_id=hr.locations.location_id
62  left join hr.countries on hr.locations.country_id=hr.countries.country_id
63  left join hr.regions on hr.countries.region_id=hr.regions.region_id
64
65  where hr.departments.department_id is null
66          or hr.employees.employee_id is null
67          or hr.employees.manager_id is null
68          or hr.locations.location_id is null
69          or hr.countries.country_id is null
70          or hr.regions.region_id is null
71  ;
72
```

```
71
72  -- 6. LET'S SEE IF KIMBERLY GRANT HAS A SALARY:
73  select salary
74  from hr.employees
75  where employee_id = 178;
76
```

SALARY
7000

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME	JOB_ID	JOB_TITLE	MANAGER_ID	LOCATION_ID	COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_NAME	HIRE_DATE
178	Grant	Kimberely	-	-	SA_REP	Sales Representative	149	-	-	-	-	-	24-MAY-07
100	King	Steven	90	Executive	AD_PRES	President	-	1700	US	United States of America	2	Americas	17-JUN-03

Download CSV

2 rows selected.

4. Dealing With Missing Data

- All Sales Representatives are from the same department, location and region and have the same manager_id

- **We can fill in all the information about K.Grant when it will be needed**

```
81
82  -- 7. LOOKING FOR SALES REPRESENTATIVES WITH MANAGER_ID = 149
83  -- TO FIND OUT ALL MISSING INFORMATION FOR KIMBERLY GRANT
84  select distinct employee_id
85          , last_name
86          , first_name
87          , hr.employees.department_id
88          , department_name
89          , hr.employees.job_id
90          , job_title
91          , hr.employees.manager_id
92          , hr.locations.location_id
93          , hr.countries.country_id
94          , country_name
95          , hr.regions.region_id
96          , region_name
97  from hr.employees
98
99  left join hr.departments
100     on hr.employees.department_id=hr.departments.department_id
101 left join hr.jobs on hr.employees.job_id=hr.jobs.job_id
102 left join hr.locations
103     on hr.departments.location_id=hr.locations.location_id
104 left join hr.countries on hr.locations.country_id=hr.countries.country_id
105 left join hr.regions on hr.countries.region_id=hr.regions.region_id
106
107 where job_title = 'Sales Representative' and hr.employees.manager_id = 149
108
109 order by employee_id
110 ;
111
```

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME	JOB_ID	JOB_TITLE	MANAGER_ID	LOCATION_ID	COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_NAME
174	Abel	Ellen	80	Sales	SA_REP	Sales Representative	149	2500	UK	United Kingdom	1	Europe
175	Hutton	Alyssa	80	Sales	SA_REP	Sales Representative	149	2500	UK	United Kingdom	1	Europe
176	Taylor	Jonathon	80	Sales	SA_REP	Sales Representative	149	2500	UK	United Kingdom	1	Europe
177	Livingston	Jack	80	Sales	SA_REP	Sales Representative	149	2500	UK	United Kingdom	1	Europe
178	Grant	Kimberely	-	-	SA_REP	Sales Representative	149	-	-	-	-	-
179	Johnson	Charles	80	Sales	SA_REP	Sales Representative	149	2500	UK	United Kingdom	1	Europe

Download CSV

6 rows selected.

5. Correcting Our Data and Making Computations

- After correction K.Grant has the department_id = 80 and the department has right number of employees

```
110
111 -- 8. CORRECTED EMPLOYEES NUMBER IN DEPARTMENTS
112 select count(distinct employee_id) as EMPLOYEES_TOTAL
113     , coalesce(department_id, 80) as Department_ID
114 from hr.employees
115 group by coalesce(department_id, 80)
116 order by department_id
117 ;
118
119
120 -- 9. SALARY DISTRIBUTION IN DEPARTMENTS
121 select count(distinct employee_id) as EMPLOYEE_TOTAL
122     , sum(salary) as TOTAL_SALARY
123     , max(salary) as MAX_SALARY
124 -- rounding average salary
125     , round(avg(salary)) as AVG_SALARY
126     , UPPER(coalesce(department_name, 'Sales')) as DEPARTMENT_NAME
127 from hr.employees
128 inner join hr.departments
129     on coalesce(hr.employees.department_id, 80) =
130     coalesce(hr.departments.department_id, 80)
131 group by department_name
132 order by total_salary desc
133 ;
134
```

AFTER CORRECTION

EMPLOYEES_TOTAL	DEPARTMENT_ID
1	10
2	20
6	30
1	40
45	50
5	60
1	70
35	80
3	90
6	100
2	110

Download CSV
11 rows selected.

BEFORE CORRECTION

EMPLOYEES_TOTAL	DEPARTMENT_ID
1	10
2	20
6	30
1	40
45	50
5	60
1	70
34	80
3	90
6	100
2	110
1	-

Download CSV
12 rows selected.

EMPLOYEE_TOTAL	TOTAL_SALARY	MAX_SALARY	AVG_SALARY	DEPARTMENT_NAME
35	311500	14000	8900	SALES
45	156400	8200	3476	SHIPPING
3	58000	24000	19333	EXECUTIVE
6	51608	12008	8601	FINANCE
5	28800	9000	5760	IT
6	24900	11000	4150	PURCHASING
2	20308	12008	10154	ACCOUNTING
2	19000	13000	9500	MARKETING
1	10000	10000	10000	PUBLIC RELATIONS
1	6500	6500	6500	HUMAN RESOURCES
1	4400	4400	4400	ADMINISTRATION

Download CSV
11 rows selected.

- We have 4 departments with much higher salary amount than other 7 departments

- **Let's take a closer look at Executive, Finance, Sales and Shipping departments**

6. Executive, Finance, Sales, Shipping Departments

- Salary in Executive, Finance, and Sales departments?
- Salary in Shipping department higher than 5000?

- **49 employees in total**

```sql
-- 10. LET'S TAKE A CLOSER LOOK ON EXECUTIVE, SALES,
--     SHIPPING AND FINANCE DEPARTMENTS

select  employee_id
-- the last and first names were joined to one column named Full_Name
    , UPPER(last_name)||' '||first_name as FULL_NAME
    , department_name
    , job_title
    , salary
-- changing the date format to YYYY
    , to_char(hire_date,'YYYY') as HIRE_DATE
-- counting rows in each department separately
    , (row_number() over (partition by department_name order by employee_id))
       as Row_Number
from hr.employees
inner join hr.departments
    on coalesce(hr.employees.department_id, 80) =
    coalesce(hr.departments.department_id, 80)
inner join hr.jobs on hr.employees.job_id = hr.jobs.job_id
where department_name = 'Executive'
    or coalesce(department_name,'Sales') = 'Sales'
    or department_name = 'Shipping' and salary >= 5000
    or department_name = 'Finance'
;
```

EMPLOYEE_TOTAL	TOTAL_SALARY	MAX_SALARY	AVG_SALARY	DEPARTMENT_NAME
35	311500	14000	8900	SALES
45	156400	8200	3476	SHIPPING
3	58000	24000	19333	EXECUTIVE
6	51608	12008	8601	FINANCE
5	28800	9000	5760	IT
6	24900	11000	4150	PURCHASING
2	20308	12008	10154	ACCOUNTING
2	19000	13000	9500	MARKETING
1	10000	10000	10000	PUBLIC RELATIONS
1	6500	6500	6500	HUMAN RESOURCES
1	4400	4400	4400	ADMINISTRATION

Download CSV

11 rows selected.

The Part of the Table

EMPLOYEE_ID	FULL_NAME	DEPARTMENT_NAME	JOB_TITLE	SALARY	HIRE_DATE	ROW_NUMBER
100	KING Steven	Executive	President	24000	2003	1
101	KOCHHAR Neena	Executive	Administration Vice President	17000	2005	2
102	DE HAAN Lex	Executive	Administration Vice President	17000	2001	3
108	GREENBERG Nancy	Finance	Finance Manager	12008	2002	1
109	FAVIET Daniel	Finance	Accountant	9000	2002	2
110	CHEN John	Finance	Accountant	8200	2005	3
111	SCIARRA Ismael	Finance	Accountant	7700	2005	4
112	URMAN Jose Manuel	Finance	Accountant	7800	2006	5
113	POPP Luis	Finance	Accountant	6900	2007	6
145	RUSSELL John	Sales	Sales Manager	14000	2004	1
146	PARTNERS Karen	Sales	Sales Manager	13500	2005	2
147	ERRAZURIZ Alberto	Sales	Sales Manager	12000	2005	3
148	CAMBRAULT Gerald	Sales	Sales Manager	11000	2007	4
149	ZLOTKEY Eleni	Sales	Sales Manager	10500	2008	5
150	TUCKER Peter	Sales	Sales Representative	10000	2005	6
151	BERNSTEIN David	Sales	Sales Representative	9500	2005	7
152	HALL Peter	Sales	Sales Representative	9000	2005	8

⋮ ⋮ ⋮

166	ANDE Sundar	Sales	Sales Representative	6400	2008	22
167	BANDA Amit	Sales	Sales Representative	6200	2008	23
168	OZER Lisa	Sales	Sales Representative	11500	2005	24
169	BLOOM Harrison	Sales	Sales Representative	10000	2006	25
170	FOX Tayler	Sales	Sales Representative	9600	2006	26
171	SMITH William	Sales	Sales Representative	7400	2007	27
172	BATES Elizabeth	Sales	Sales Representative	7300	2007	28
173	KUMAR Sundita	Sales	Sales Representative	6100	2008	29
174	ABEL Ellen	Sales	Sales Representative	11000	2004	30
175	HUTTON Alyssa	Sales	Sales Representative	8800	2005	31
176	TAYLOR Jonathon	Sales	Sales Representative	8600	2006	32
177	LIVINGSTON Jack	Sales	Sales Representative	8400	2006	33
178	GRANT Kimberely	Sales	Sales Representative	7000	2007	34
179	JOHNSON Charles	Sales	Sales Representative	6200	2008	35
120	WEISS Matthew	Shipping	Stock Manager	8000	2004	1
121	FRIPP Adam	Shipping	Stock Manager	8200	2005	2
122	KAUFLING Payam	Shipping	Stock Manager	7900	2003	3
123	VOLLMAN Shanta	Shipping	Stock Manager	6500	2005	4
124	MOURGOS Kevin	Shipping	Stock Manager	5800	2007	5

Download CSV

49 rows selected.

- We see K.Grant in our results

7-8-9. Employees To Be Laid Off

- 11 employees to be laid off
- It is 10.3% of total employees number
- Savings per month will be 83.5 thousands
- Savings per year will be a little bit more than 1 million

```sql
-- 11. EMPLOYEES TO BE LAID OFF
-- Creating view to use it later, when needed
create view  empl_laid_off as
select distinct employee_id
    , UPPER(last_name)||' '||first_name as FULL_NAME
    , hr.departments.department_id
    , job_title
    , salary
    , to_char(hire_date,'YYYY') as HIRE_DATE
-- Writing full email addresses
    , LOWER(email)||coalesce (null, '@jb.com ') as EMAIL_ADDRESS
from hr.employees
inner join hr.departments
    on coalesce(hr.employees.department_id, 80) =
    coalesce(hr.departments.department_id, 80)
inner join hr.jobs on hr.employees.job_id = hr.jobs.job_id
where department_name = 'Executive' and to_char(hire_date,'YYYY') = '2001'
    or department_name = 'Finance' and salary <7800
    or department_name = 'Shipping' and salary > 5000 and salary < 8000
    or job_title = 'Sales Representative' and salary <= 7000
       and to_char(hire_date,'YYYY')  = '2008'
;

-- Looking at the created view
select *
from empl_laid_off
order by employee_id, department_id
;
```

EMPLOYEE_ID	FULL_NAME	DEPARTMENT_ID	JOB_TITLE	SALARY	HIRE_DATE	EMAIL_ADDRESS
102	DE HAAN Lex	90	Administration Vice President	17000	2001	ldehaan@jb.com
111	SCIARRA Ismael	100	Accountant	7700	2005	isciarra@jb.com
113	POPP Luis	100	Accountant	6900	2007	lpopp@jb.com
122	KAUFLING Payam	50	Stock Manager	7900	2003	pkauflin@jb.com
123	VOLLMAN Shanta	50	Stock Manager	6500	2005	svollman@jb.com
124	MOURGOS Kevin	50	Stock Manager	5800	2007	kmourgos@jb.com
165	LEE David	80	Sales Representative	6800	2008	dlee@jb.com
166	ANDE Sundar	80	Sales Representative	6400	2008	sande@jb.com
167	BANDA Amit	80	Sales Representative	6200	2008	abanda@jb.com
173	KUMAR Sundita	80	Sales Representative	6100	2008	skumar@jb.com
179	JOHNSON Charles	80	Sales Representative	6200	2008	cjohnson@jb.com

Download CSV
11 rows selected.

```sql
-- 12. SAVINGS PER MONTH AND YEAR
-- Using the view from 11th statement
select count(distinct employee_id) as LAID_OFF_TOTAL
    , sum(salary) AS SAVINGS_PER_MONTH
    , sum(salary)*12 as SAVINGS_PER_YEAR
    , round(((count(employee_id)/107)*100),1) as PERCENT_OF_TOTAL_EMPLOYEES
from empl_laid_off
;
```

LAID_OFF_TOTAL	SAVINGS_PER_MONTH	SAVINGS_PER_YEAR	PERCENT_OF_TOTAL_EMPLOYEES
11	83500	1002000	10.3

Download CSV

```sql
-- 13. EMAILS TO EMPLOYEES WHO WILL BE LAID OFF
-- Using the view from 11th statement
select full_name
    , email_address
from empl_laid_off
;


-- 14. SENT EMAILS
-- Using the view from 11th statement
select full_name
-- Adding an empty column to use later (when an email is sent)
    , coalesce (null, ' ') as EMAIL_SENT
from empl_laid_off
;
```

FULL_NAME	EMAIL_ADDRESS
DE HAAN Lex	ldehaan@jb.com
ANDE Sundar	sande@jb.com
MOURGOS Kevin	kmourgos@jb.com
POPP Luis	lpopp@jb.com
LEE David	dlee@jb.com
BANDA Amit	abanda@jb.com
KAUFLING Payam	pkauflin@jb.com
JOHNSON Charles	cjohnson@jb.com
SCIARRA Ismael	isciarra@jb.com
KUMAR Sundita	skumar@jb.com
VOLLMAN Shanta	svollman@jb.com

Download CSV
11 rows selected.

FULL_NAME	EMAIL_SENT
DE HAAN Lex	
ANDE Sundar	
MOURGOS Kevin	
POPP Luis	
LEE David	
BANDA Amit	
KAUFLING Payam	
JOHNSON Charles	
SCIARRA Ismael	
KUMAR Sundita	
VOLLMAN Shanta	

Download CSV
11 rows selected.

- Emails list with full email addresses to inform employees
- Empty column – to mark ✓ when email is sent

10. Employees Relocation / Additional Office

- Sales and Shipping Departments have the highest number of employees (80 in total)
- Sales Department located in Oxford, UK
- Shipping Department located in South San Francisco, USA
- 17 employees from Sales and 13 employees from Shipping Departments (28% of total in both departments) with the salary higher than an average must be relocated into other cities in the UK and USA with lower leasing costs

```sql
-- 1. DEPARTMENTS BY LOCATION
select distinct employee_id
    , hr.departments.department_id
    , department_name
    , hr.departments.location_id
    , country_name, city
    , hr.countries.country_id
    , region_id
    , salary
from hr.employees
full outer join hr.departments
    on coalesce(hr.departments.department_id, 80) =
        coalesce(hr.employees.department_id,80)
full outer join hr.locations
    on hr.departments.location_id = hr.locations.location_id
full outer join hr.countries
    on hr.locations.country_id = hr.countries.country_id
where department_name = 'Shipping'
    or coalesce(department_name, 'Sales') = 'Sales'
    and employee_id is NOT NULL and salary is NOT NULL
order by department_name
;

-- 2. EMPLOYEES WITH SALARY HIGHER THAN THE AVERAGE
--    IN SALES AND SHIPPING DEPARTMENTS
select distinct employee_id
    , hr.departments.department_id
    , department_name
    , hr.departments.location_id
    , country_name, city
    , hr.countries.country_id
    , region_id
    , salary
from hr.employees
full outer join hr.departments
    on hr.departments.department_id = hr.employees.department_id
full outer join hr.locations
    on hr.departments.location_id = hr.locations.location_id
full outer join hr.countries
    on hr.locations.country_id = hr.countries.country_id
where coalesce(department_name, 'Sales') = 'Sales' and salary >= (
        select round(avg(salary)) as AVG_SALARY
        from hr.employees
         where coalesce(hr.employees.department_id, 80) = 80
         )
    or
     department_name = 'Shipping' and salary > (
        select round(avg(salary)) as AVG_SALARY
        from hr.employees
        where hr.employees.department_id = 50
    )
order by department_name, salary desc
;
```

EMPLOYEE_ID	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	COUNTRY_NAME	CITY	COUNTRY_ID	REGION_ID	SALARY
145	80	Sales	2500	United Kingdom	Oxford	UK	1	14000
146	80	Sales	2500	United Kingdom	Oxford	UK	1	13500
147	80	Sales	2500	United Kingdom	Oxford	UK	1	12000
168	80	Sales	2500	United Kingdom	Oxford	UK	1	11500
148	80	Sales	2500	United Kingdom	Oxford	UK	1	11000
174	80	Sales	2500	United Kingdom	Oxford	UK	1	11000
149	80	Sales	2500	United Kingdom	Oxford	UK	1	10500
162	80	Sales	2500	United Kingdom	Oxford	UK	1	10500
150	80	Sales	2500	United Kingdom	Oxford	UK	1	10000
156	80	Sales	2500	United Kingdom	Oxford	UK	1	10000
169	80	Sales	2500	United Kingdom	Oxford	UK	1	10000
170	80	Sales	2500	United Kingdom	Oxford	UK	1	9600
151	80	Sales	2500	United Kingdom	Oxford	UK	1	9500
157	80	Sales	2500	United Kingdom	Oxford	UK	1	9500
163	80	Sales	2500	United Kingdom	Oxford	UK	1	9500
152	80	Sales	2500	United Kingdom	Oxford	UK	1	9000
158	80	Sales	2500	United Kingdom	Oxford	UK	1	9000
121	50	Shipping	1500	United States of America	South San Francisco	US	2	8200
120	50	Shipping	1500	United States of America	South San Francisco	US	2	8000
122	50	Shipping	1500	United States of America	South San Francisco	US	2	7900
123	50	Shipping	1500	United States of America	South San Francisco	US	2	6500
124	50	Shipping	1500	United States of America	South San Francisco	US	2	5800
184	50	Shipping	1500	United States of America	South San Francisco	US	2	4200
185	50	Shipping	1500	United States of America	South San Francisco	US	2	4100
192	50	Shipping	1500	United States of America	South San Francisco	US	2	4000
193	50	Shipping	1500	United States of America	South San Francisco	US	2	3900
188	50	Shipping	1500	United States of America	South San Francisco	US	2	3800
137	50	Shipping	1500	United States of America	South San Francisco	US	2	3600
189	50	Shipping	1500	United States of America	South San Francisco	US	2	3600
141	50	Shipping	1500	United States of America	South San Francisco	US	2	3500

Download CSV

30 rows selected.

Conclusions

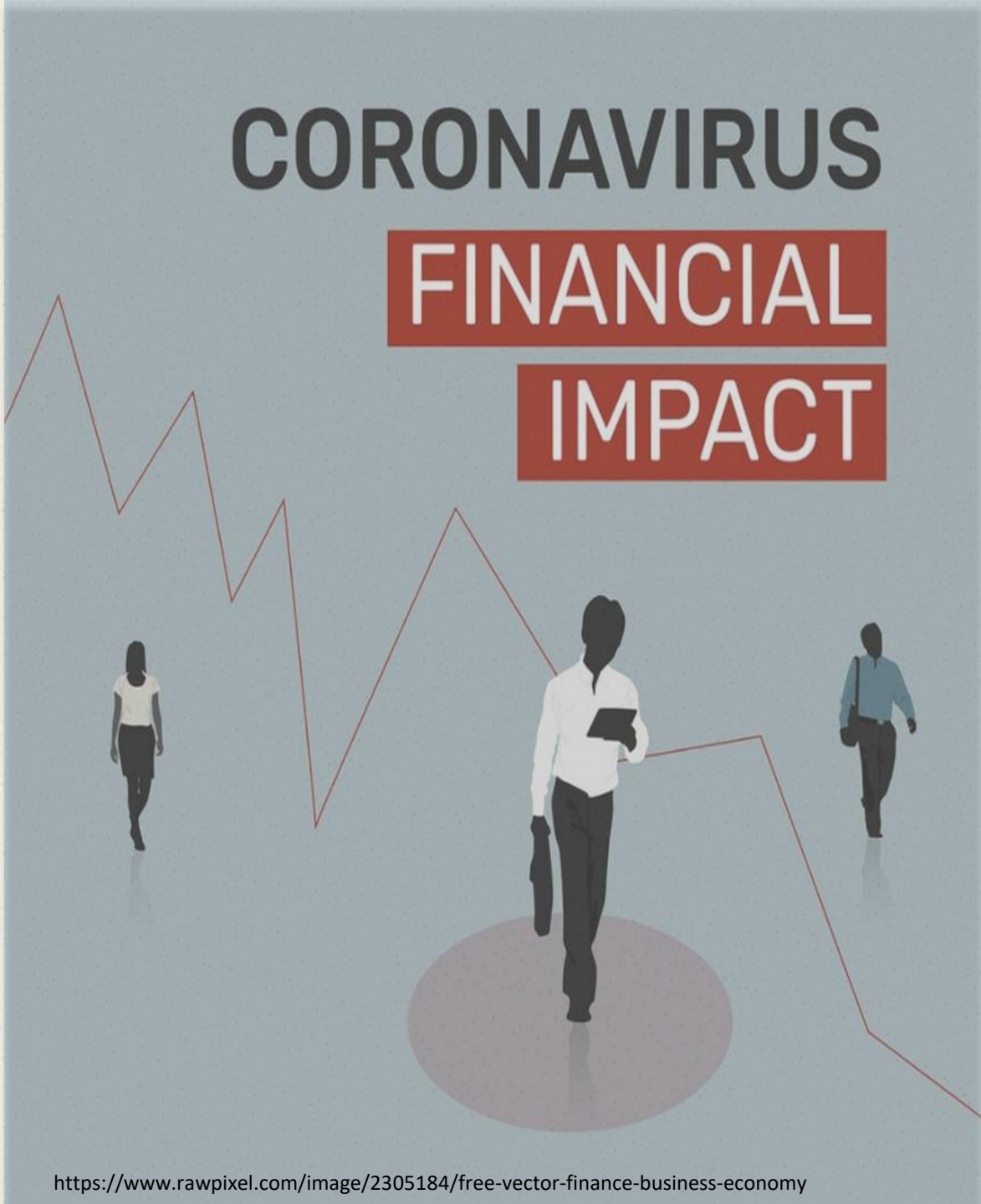
1. **LAY OFF THE PART OF THE EMPLOYEES**
   for savings to survive this hard COVID-19 time

   - **11 employees** from **4 departments** to be laid off
   - **10.3%** of total employees number
   - Savings per month - **83.5 thousands**
   - Savings per year - **a little bit more than 1 million**

2. **RELOCATE THE PART OF THE EMPLOYEES**
   from departments with the highest number of employees

   - **17 employees from Sales and 13 employees from Shipping Departments** (**28%** in both departments) must be relocated into other cities in the UK and USA with lower office leasing costs

‼ For additional questions on how I arrived on this conclusions, please see my GitHub account:
https://github.com/OlshIna/Oracle-SQL-project



https://www.rawpixel.com/image/2305184/free-vector-finance-business-economy