

BDM PROJECT

Dataset

With reference to employee payroll, this dataset provides crucial information.

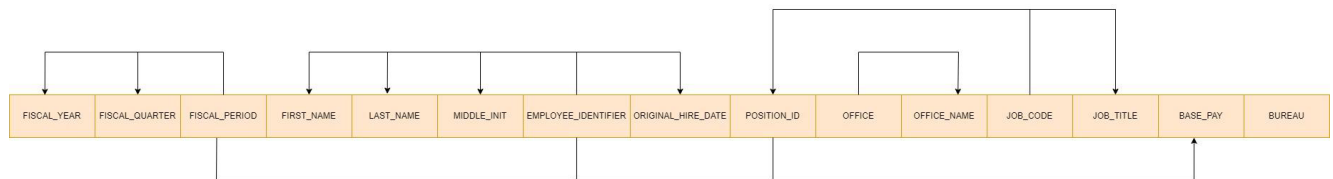
Particularly, The dataset focuses on the payroll details of employees for the fiscal years of 2016, 2017 and 2018 with payroll being given out every fiscal quarter.

In order to properly manage their finances and make sure that employees are paid fairly and on time, businesses may benefit greatly from this information.

Data source: <https://catalog.data.gov/dataset/employee-payroll>

Data Normalization

Raw data before normalization:



- The data is redundant and has created the following anomalies:
 - **Creation Anomalies:**
 - If a new employee is added, and the employee's office or bureau is not already present in the respective tables, then there would be a creation anomaly.
 - **Updation Anomalies:**
 - If the name of an office is updated, then all the records that reference that office name would need to be updated to avoid inconsistency.
 - **Deletion Anomalies:**
 - If the bureau of payroll is deleted, then all the records related to the bureau would be deleted, which would result in losing crucial information on other columns.

Normalized data structure:

<u>PAYROLL_ID</u>	E_ID	FISCAL_PERIOD	POSITION_ID	BASE_PAY
-------------------	------	---------------	-------------	----------

<u>E_ID</u>	E_FNAME	E_LNAME	E_MIDDLE_INIT	E_HIRE_DATE
-------------	---------	---------	---------------	-------------

<u>FISCAL_PERIOD</u>	FISCAL_YEAR	FISCAL_QUATER
----------------------	-------------	---------------

<u>BUREAU_ID</u>	BUREAU_NAME
------------------	-------------

<u>JOB_CODE</u>	JOB_TITLE
-----------------	-----------

<u>OFFICE_ID</u>	OFFICE_NAME	BUREAU_ID
------------------	-------------	-----------

<u>POSITION_ID</u>	JOB_CODE
--------------------	----------

<u>E_OFC_ID</u>	E_ID	OFFICE_ID
-----------------	------	-----------

- **EMPLOYEE**
 - PK: E_ID
 - E_ID determines E_FNAME, E_LNAME, E_MIDDLE_INIT and E_HIRE_DATE
- **FISCAL_YEAR**
 - PK: FISCAL_PERIOD
 - FISCAL_PERIOD determines FISCAL_YEAR and FISCAL_QUATER
- **BUREAU**
 - PK: BUREAU_ID
 - BUREAU_ID determines BUREAU_NAME
- **JOB**
 - PK: JOB_CODE
 - JOB_CODE determines JON_TITLE
- **OFFICE**
 - PK: OFFICE_ID
 - FK: BUREAU_ID
 - OFFICE_ID determines OFFICE_TITLE

- **POSITION**
 - PK: POSITION_ID
 - FK: JOB_CODE
- **EMP_OFFICE**
 - PK: E_OFC_ID
 - FK1: E_ID
 - FK2: OFFICE_ID
- **PAYROLL**
 - PK: PAYROLL_ID
 - FK1: E_ID
 - FK2: FISCAL_PERIOD
 - FK3: POSITION_ID
 - FK4: BASE_BAY

Entity-Relation Diagram

Business Rules:

One Employee is assigned to Many Job.

One Job contains Many Position.

One Position is allotted Many Payroll.

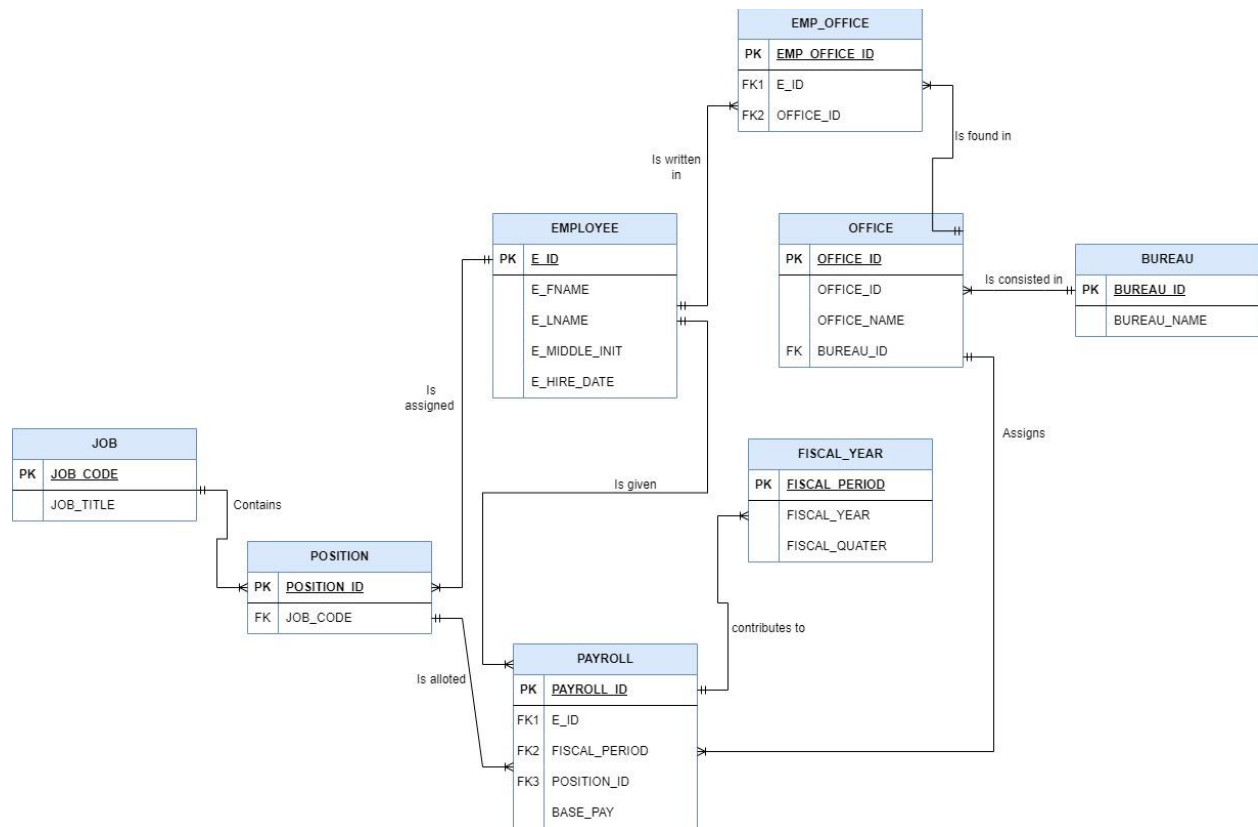
Many Employees have Many Jobs.

Many Financial years contributes to One Payroll.

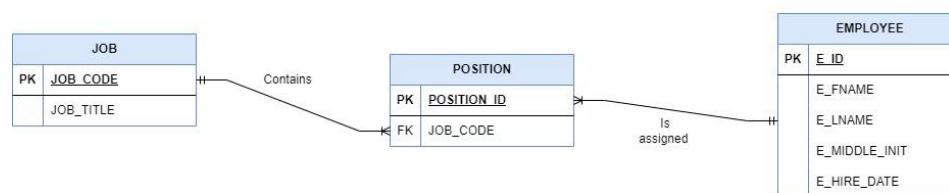
Many Employees work in Many Offices.

Many Offices are consisted in One Bureau.

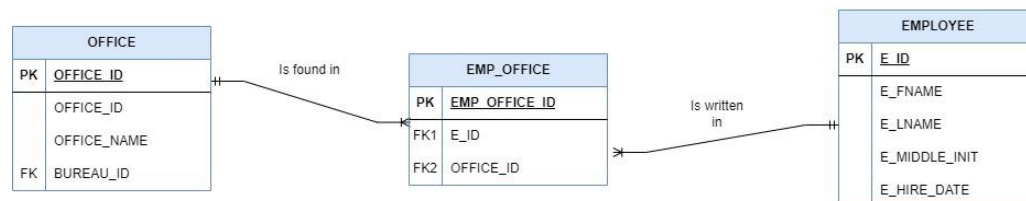
ER Diagram



Employee and Job have many to many relationship which is bridged by the Position table.



Employee and Office have many to many relationship which is bridged by EMP_OFFICE table.



SQL Queries

1. How many employees are employed at the company?

```
SELECT COUNT(DISTINCT E_ID)
AS unique_employee
FROM PAYROLL
```

Explanation:

- This counts the number of unique ID numbers in the payroll table.
- In knowing the number of unique numbers, and excluding duplicates, you can see the number of distinct employees working and getting paid.

Output:

```
SELECT COUNT(DISTINCT E_ID)
AS unique_employee
FROM PAYROLL
```

UNIQUE_EMPLOYEE	
11	

2. What is the average Base Pay for employees whose name is “Pamela”?

```
SELECT AVG(BASE_PAY)
FROM PAYROLL
JOIN EMPLOYEE ON PAYROLL.E_ID = EMPLOYEE.E_ID
WHERE E_FNAME = 'PAMELA'
```

Explanation:

- This joins the two foreign keys to link two different tables.
- The WHERE clause filter for Pamela only, and excludes all other individuals in the data set.

Output:

```
SELECT AVG(BASE_PAY)
FROM PAYROLL
JOIN EMPLOYEE ON PAYROLL.E_ID = EMPLOYEE.E_ID
WHERE E_FNAME = 'PAMELA'
```

AVG(BASE_PAY)
15141

3. What is the average pay of employees who are paid more than \$10,000?

```
SELECT ROUND(AVG(BASE_PAY),2) AS BASE_PAY
FROM PAYROLL
WHERE BASE_PAY > 10000
```

Explanation:

- This is important because it will indicate what the top percentage of employee's are earning.
- This information is important for employers to understand when awarding raises, and promotions.

```
SELECT ROUND(AVG(BASE_PAY),2) AS BASE_PAY
FROM PAYROLL
WHERE BASE_PAY > 10000
```

BASE_PAY
16892.7

4. What employee has the highest salary?

```
SELECT E_FNAME, E_LNAME
FROM EMPLOYEE
NATURAL JOIN PAYROLL
WHERE BASE_PAY = (SELECT MAX(BASE_PAY) FROM PAYROLL)
```

Explanation:

- The inner query of SELECT MAX filters to the employee with the highest salary.
- Then joining the two tables allows the tables to form a relationship. The end result gives the first and last name of the employee with the highest salary.

Output:

```
SELECT E_FNAME, E_LNAME
FROM EMPLOYEE
NATURAL JOIN PAYROLL
WHERE BASE_PAY = (SELECT MAX(BASE_PAY) FROM PAYROLL)
```

E_FNAME	E_LNAME
AMRITH	AAKRE

5. Show all the employees with last name 'Pamela'

```
SELECT * FROM EMPLOYEE
WHERE E_FNAME LIKE 'PAMELA'
```

Explanation:

- Use like clauses to compare strings, and find employees that have last name Pamela.

```
SELECT * FROM EMPLOYEE
WHERE E_FNAME LIKE 'PAMELA'
```

E_ID	E_FNAME	E_LNAME	E_MIDDLE_INIT	E_HIRE_DATE
11	PAMELA	HUNT	J	11/01/1988
9	PAMELA	HUMPHRIES	C	11/18/2002
10	PAMELA	WILLIAMSON	R	04/17/2006

6. Show all the employees that work in 'Stroger Hospital of Cook City'

```
SELECT * FROM EMPLOYEE
WHERE EXISTS
(SELECT OFFICE_ID FROM EMP_OFFICE
WHERE OFFICE_ID IN
(SELECT OFFICE_ID FROM OFFICE
WHERE OFFICE_NAME ='STROGER HOSPITAL OF COOK CNTY'))
```

Explanation:

- We use nested select from employee to emp_office, from emp_office to office, to find out which employees work in 'Stroger Hospital of Cook City'.

Output:

```

1  SELECT * FROM EMPLOYEE
2  WHERE EXISTS
3  (SELECT OFFICE_ID FROM EMP_OFFICE
4  WHERE OFFICE_ID IN
5  (SELECT OFFICE_ID FROM OFFICE
6  WHERE OFFICE_NAME = 'STROGER HOSPITAL OF COOK CNTY'))
7

```

E_ID	E_FNAME	E_LNAME	E_MIDDLE_INIT	E_HIRE_DATE
4	TRACY	WILLIAMS	-	08/13/2001
11	PAMELA	HUNT	J	11/01/1988
1	AMRITH	AAKRE	K	07/16/2005
6	ZULEYMA	RIVERA-FARIAS	V	08/31/1998
8	ABDALLAH	ABDELHAMID	F	07/27/2011
9	PAMELA	HUMPHRIES	C	11/18/2002
7	KHALIL	ABBOUD	E	12/17/1982
2	TRESA	REED-HUMPHRIES	C	04/28/2008
3	RUTH	ABARCA	A	07/06/2016
10	PAMELA	WILLIAMSON	R	04/17/2006

7. Show list of employees who work in Bureau of Health

```

SELECT E_ID, E_FNAME, E_LNAME
FROM EMPLOYEE
WHERE E_ID IN (SELECT E_ID FROM EMP_OFFICE WHERE OFFICE_ID IN
(SELECT OFFICE_ID FROM OFFICE WHERE BUREAU_ID IN (SELECT
BUREAU_ID FROM BUREAU
WHERE BUREAU.BUREAU_NAME = 'Bureau of Health')));

```

Explanation:

- We use nested query from employee to office to bureau to get employees that work in Bureau of health.

Output:

```

SELECT E_ID, E_FNAME, E_LNAME
FROM EMPLOYEE
WHERE E_ID IN (SELECT E_ID FROM EMP_OFFICE WHERE OFFICE_ID IN (SELECT OFFICE_ID FROM OFFICE WHERE BUREAU_ID IN (SELECT BUREAU_ID FROM BUREAU
WHERE BUREAU.BUREAU_NAME = 'Bureau of Health'))))

```

E_ID	E_FNAME	E_LNAME
4	TRACY	WILLIAMS
6	ZULEYMA	RIVERA-FARIAS

8. Display the job id for employees who have base pay between 10000 to 30000

```

SELECT JOB_CODE FROM JOB

```



```

JOIN POSITION USING(JOB_CODE)
JOIN PAYROLL USING(POSITION_ID)
WHERE BASE_PAY>=10000 AND BASE_PAY<=30000

```

Explanation:

- Join table job to position and position to payroll table.
- We compare base pay using the conditional operator AND.

Output:

```

SELECT JOB_CODE FROM JOB
JOIN POSITION USING(JOB_CODE)
JOIN PAYROLL USING(POSITION_ID)
WHERE BASE_PAY>=10000 AND BASE_PAY<=30000

```

JOB_CODE
1172
1172
1172

9. Show employees, their job id, position id and their max base pay by office

```

SELECT
E.E_ID,PO.JOB_CODE,P.POSITION_ID,O.OFFICE_NAME,P.BASE_PAY
FROM EMPLOYEE E
INNER JOIN PAYROLL P ON E.E_ID = P.E_ID
INNER JOIN POSITION PO ON PO.POSITION_ID = P.POSITION_ID
INNER JOIN JOB J ON J.JOB_CODE = PO.JOB_CODE
INNER JOIN EMP_OFFICE EO ON EO.E_ID = E.E_ID
INNER JOIN OFFICE O ON O.OFFICE_ID = EO.OFFICE_ID
INNER JOIN
(
    SELECT OFFICE_NAME, MAX(BASE_PAY) AS MAX_BASE_PAY
    FROM PAYROLL P
    INNER JOIN EMP_OFFICE EO ON EO.E_ID = P.E_ID
    INNER JOIN OFFICE O ON O.OFFICE_ID = EO.OFFICE_ID
    GROUP BY OFFICE_NAME
)
MAX_PAY ON MAX_PAY.OFFICE_NAME = O.OFFICE_NAME AND
P.BASE_PAY = MAX_PAY.MAX_BASE_PAY;

```

Explanation:

- We have used inner join to join employee and payroll, payroll to employee, employee to job, employee to emp_office, emp_office to office.
- We have also done inner join on nested query via payroll, from which we join emp_office, and emp_office to office.
- In the nested query, we aggregated the data by office.
- From this we have derived max base pay of employees for each office through the payroll table.

Output:

```

SELECT E.E_ID,PO.JOB_CODE,P.POSITION_ID,O.OFFICE_NAME,P.BASE_PAY
FROM EMPLOYEE E
INNER JOIN PAYROLL P ON E.E_ID = P.E_ID
INNER JOIN POSITION PO ON PO.POSITION_ID = P.POSITION_ID
INNER JOIN JOB J ON J.JOB_CODE = PO.JOB_CODE
INNER JOIN EMP_OFFICE EO ON EO.E_ID = E.E_ID
INNER JOIN OFFICE O ON O.OFFICE_ID = EO.OFFICE_ID
INNER JOIN
(
    SELECT OFFICE_NAME, MAX(BASE_PAY) AS MAX_BASE_PAY
    FROM PAYROLL P
    INNER JOIN EMP_OFFICE EO ON EO.E_ID = P.E_ID
    INNER JOIN OFFICE O ON O.OFFICE_ID = EO.OFFICE_ID
    GROUP BY OFFICE_NAME
)
MAX_PAY ON MAX_PAY.OFFICE_NAME = O.OFFICE_NAME AND P.BASE_PAY = MAX_PAY.MAX_BASE_PAY;

```

E_ID	JOB_CODE	POSITION_ID	OFFICE_NAME	BASE_PAY
9	1942	1000415	Cermak Health Services	23775.8
6	4688	1400686	STROGER HOSPITAL OF COOK CNTY	13772.62
3	79	1600398	DEPT. OF HUMAN RESOURCES	7344.6
1	1172	9510200	STATES ATTORNEY	24473.38
5	1564	9512640	ADULT PROBATION DEPT.	24006.53
2	5742	9515112	CLERK OF CRCT CRT OFF. OF CLER	14882

10. What is the base pay of employees that have middle initial

```

SELECT EMPLOYEE.E_ID, EMPLOYEE.E_FNAME,
EMPLOYEE.E_MIDDLE_INIT, PAYROLL.BASE_PAY
FROM EMPLOYEE
JOIN PAYROLL ON EMPLOYEE.E_ID=PAYROLL.E_ID
WHERE E_MIDDLE_INIT IS NOT NULL;

```

Explanation:

- We join employee and payroll to find the base pay.
- We use IS NOT NULL function to filter the employees that have middle names.

Output:

```
SELECT EMPLOYEE.E_ID, EMPLOYEE.E_FNAME, EMPLOYEE.E_MIDDLE_INIT, PAYROLL.BASE_PAY
FROM EMPLOYEE
JOIN PAYROLL ON EMPLOYEE.E_ID=PAYROLL.E_ID
WHERE E_MIDDLE_INIT IS NOT NULL;
```

E_ID	E_FNAME	E_MIDDLE_INIT	BASE_PAY
1	AMRITH	K	20088
1	AMRITH	K	23436
1	AMRITH	K	20422.82
1	AMRITH	K	23904.8
1	AMRITH	K	20745.8
1	AMRITH	K	24473.38
1	AMRITH	K	21217.35

11. Who is the senior most employee?

```
SELECT E_FNAME,
E_LNAME,
E_HIRE_DATE FROM EMPLOYEE
WHERE E_HIRE_DATE=(SELECT MIN(E_HIRE_DATE) FROM EMPLOYEE);
```

Explanation:

- The query checks for entity with minimum hire date (Using MIN())

Output:

```
SELECT E_FNAME,
E_LNAME,
E_HIRE_DATE FROM EMPLOYEE
WHERE E_HIRE_DATE=(SELECT MIN(E_HIRE_DATE) FROM EMPLOYEE);
```

E_FNAME	E_LNAME	E_HIRE_DATE
KHALIL	ABBOUD	12/17/1982

12. List employees (with their full names) having more than one last name or no middle name.

```
SELECT CONCAT(CONCAT(E_FNAME, ' '),
CONCAT(CONCAT(E_MIDDLE_INIT, ' '), E_LNAME)) AS E_FULL_NAME
FROM EMPLOYEE
```

```
WHERE E_LNAME LIKE '%-%' OR E_MIDDLE_INIT IS NULL  
ORDER BY E_FNAME;
```

Explanation:

- In order to know if any of the employees have two last names, we are checking for '-' in between the names
- To get employees without a middle name, we are checking for NULL in the middle name column.
- We are using nested CONCAT() to get the full name of the employee.

```
SELECT CONCAT(CONCAT(E_FNAME, ' '),  
CONCAT(CONCAT(E_MIDDLE_INIT, ' '), E_LNAME)) AS E_FULL_NAME  
FROM EMPLOYEE  
WHERE E_LNAME LIKE '%-%' OR E_MIDDLE_INIT IS NULL  
ORDER BY E_FNAME;
```

E_FULL_NAME
TRACY WILLIAMS
TRES'SA C REED-HUMPHRIES
ZULEYMA V RIVERA-FARIAS

13. List the names of employees who got more than one payroll for different positions during the same fiscal period

```
SELECT DISTINCT E1.E_ID,  
CONCAT(CONCAT(E1.E_FNAME, ' '), E1.E_LNAME) AS E_NAME  
FROM PAYROLL P1  
JOIN PAYROLL P2 ON P1.E_ID = P2.E_ID AND P1.FISCAL_PERIOD =  
P2.FISCAL_PERIOD AND P1.POSITION_ID <> P2.POSITION_ID  
JOIN EMPLOYEE E1 ON P1.E_ID = E1.E_ID  
JOIN EMPLOYEE E2 ON P2.E_ID = E2.E_ID  
ORDER BY E_NAME;
```

Explanation:

- We are doing a **SELF JOIN** on the Payroll table for retrieving entities with similar employee id and fiscal period.
- We are then doing JOINS with the Employee tables to get the names of the employees.

Output:

```

SELECT DISTINCT E1.E_ID,
CONCAT(CONCAT(E1.E_FNAME, ' '), E1.E_LNAME) AS E_NAME
FROM PAYROLL P1
JOIN PAYROLL P2 ON P1.E_ID = P2.E_ID AND
P1.FISCAL_PERIOD = P2.FISCAL_PERIOD AND P1.POSITION_ID <> P2.POSITION_ID
JOIN EMPLOYEE E1 ON P1.E_ID = E1.E_ID
JOIN EMPLOYEE E2 ON P2.E_ID = E2.E_ID
ORDER BY E_NAME;

```

E_ID	E_NAME
9	PAMELA HUMPHRIES
6	ZULEYMA RIVERA-FARIAS

14. List employee details and total pay details for those who have base pay higher than 100k for the fiscal year of 2016. Include base pay in the details returned.

```

SELECT E_FNAME, E_LNAME, BASE_PAY_2016
FROM (
    SELECT P.E_ID, ROUND(SUM(BASE_PAY), 2) AS BASE_PAY_2016
    FROM PAYROLL P
    JOIN EMPLOYEE E ON P.E_ID = E.E_ID
    JOIN FISCAL_YEAR Y ON (P.FISCAL_PERIOD = Y.FISCAL_PERIOD AND
Y.FISCAL_YEAR = '2016')
    GROUP BY P.E_ID
    HAVING SUM(BASE_PAY) >= 80000
) A
JOIN (
    SELECT E_ID, E_FNAME, E_LNAME
    FROM EMPLOYEE
) B
USING (E_ID)
ORDER BY E_FNAME;

```

Explanation:

- First SELECT
 - We are joining payroll with the employee table to get the respective employee details, then with Fiscal Year to filter the year 2016.
 - At last, we are filtering out employees with total pay (using SUM() function) greater than 80k.
- Second SELECT

- This returns the Employee details from employee table
- JOIN
 - The two tables are joined using Employee Id.

Output:

```

SELECT E_FNAME, E_LNAME, BASE_PAY_2016
FROM (
  SELECT P.E_ID, ROUND(SUM(BASE_PAY), 2) AS BASE_PAY_2016
  FROM PAYROLL P
  JOIN EMPLOYEE E ON P.E_ID = E.E_ID
  JOIN FISCAL_YEAR Y ON (P.FISCAL_PERIOD = Y.FISCAL_PERIOD AND Y.FISCAL_YEAR = '2016')
  GROUP BY P.E_ID
  HAVING SUM(BASE_PAY) >= 80000
) A
JOIN (
  SELECT E_ID, E_FNAME, E_LNAME
  FROM EMPLOYEE
) B
USING (E_ID) ORDER BY E_FNAME;

```

E_FNAME	E_LNAME	BASE_PAY_2016
AMRITH	AAKRE	87851.62
PAMELA	HUMPHRIES	84460.27
TRACY	ROBINSON	85288.75

15. List employees who are clinical nurses or clerks.

```

SELECT DISTINCT E_FNAME, E_LNAME
FROM EMPLOYEE
JOIN PAYROLL USING (E_ID)
WHERE POSITION_ID IN (
  SELECT POSITION_ID
  FROM POSITION
  WHERE JOB_CODE IN (
    SELECT JOB_CODE
    FROM JOB
    WHERE JOB_TITLE LIKE '%Clinical Nurse%' OR JOB_TITLE LIKE
    '%Clerk%'
  )
);

```

Explanation:

- Nested SELECT subqueries are used to get job codes like Clinical Nurse, or Clerk.
- Then position ids associated with the returned job ids are retrieved.

- And finally, the payroll details associated with the returned position ids are retrieved.

Output:

```
SELECT DISTINCT E_FNAME, E_LNAME
FROM EMPLOYEE
JOIN PAYROLL USING (E_ID)
WHERE POSITION_ID IN (
  SELECT POSITION_ID
  FROM POSITION
  WHERE JOB_CODE IN (
    SELECT JOB_CODE
    FROM JOB
    WHERE JOB_TITLE LIKE '%Clinical Nurse%' OR JOB_TITLE LIKE '%Clerk%'
  )
);
```

E_FNAME	E_LNAME
ZULEYMA	RIVERA-FARIAS
PAMELA	HUMPHRIES

16. Show E_ID, First name, and Last name of employees who belong to the Sheriff Bureau.

```
SELECT e.E_id, e.E_FNAME, e.E_LNAME FROM EMPLOYEE e
JOIN EMP_OFFICE eo ON e.E_ID = eo.E_ID
JOIN OFFICE o ON o.OFFICE_ID = eo.OFFICE_ID
JOIN BUREAU b ON b.BUREAU_ID = o.BUREAU_ID
Where BUREAU_NAME = 'Sheriff';
```

Explanation:

- Here we are finding a list of Employees working in the Sheriff Bureau. We have used multiple joins to join three tables.

Output:

```

1  SELECT e.E_id, e.E_FNAME, e.E_LNAME
2  FROM EMPLOYEE e
3  JOIN EMP_OFFICE eo ON e.E_ID = eo.E_ID
4  JOIN OFFICE o ON o.OFFICE_ID = eo.OFFICE_ID
5  JOIN BUREAU b ON b.BUREAU_ID = o.BUREAU_ID
6  Where BUREAU_NAME = 'Sheriff';

```

E_ID	E_FNAME	E_LNAME
8	ABDALLAH	ABDELHAMID
10	PAMELA	WILLIAMSON

17. Find the Position ID of employees who worked in the 4th Quarter of 2017.

Select po.POSITION_ID from POSITION po JOIN PAYROLL pa on
 po.POSITION_ID = pa.POSITION_ID
 JOIN FISCAL_YEAR f on f.FISCAL_PERIOD = pa.FISCAL_PERIOD where
 f.FISCAL_YEAR= 2017 AND f.FISCAL_QUATER = 4;

Explanation:

- Here, we are getting a list of Position IDs working in the 4th Quarter of 2017. We have used multiple Join to get the result.

Output:

```

1  Select po.POSITION_ID from POSITION po
2  JOIN PAYROLL pa on po.POSITION_ID = pa.POSITION_ID
3  JOIN FISCAL_YEAR f on f.FISCAL_PERIOD = pa.FISCAL_PERIOD
4  where f.FISCAL_YEAR= 2017 AND f.FISCAL_QUATER = 4;

```

POSITION_ID
9515112
9512640
1400686
600125

4 rows returned in 0.35 seconds [Download](#)

18. Display names of employees who were hired after 2000.

SELECT E_FNAME, E_LNAME, E_HIRE_DATE
 FROM EMPLOYEE
 WHERE EXTRACT (YEAR from E_HIRE_DATE)> 2000;

Explanation:

- Here we are filtering employees based on their Hiring date.

Output:


```
SELECT E_FNAME, E_LNAME, E_HIRE_DATE
FROM EMPLOYEE
WHERE EXTRACT (YEAR from E_HIRE_DATE) > 2000;
```

Results Explain Describe Saved SQL History		
E_FNAME	E_LNAME	E_HIRE_DATE
AMRITH	AAKRE	06/16/2005
TRESSA	REED-HUMPHRIES	04/28/2008
RUTH	ABARCA	06/03/2016
TRACY	WILLIAMS	08/13/2001
ABDALLAH	ABDELHAMID	06/27/2011
PAMELA	HUMPHRIES	11/18/2002
PAMELA	WILLIAMSON	04/17/2006

7 rows returned in 0.01 seconds [Download](#)

19. Find the name of the highest-paid employee/employee with the highest base salary.

```
SELECT e.E_FNAME, e.E_MIDDLE_INIT, e.E_LNAME, pa.BASE_PAY
FROM employee e
JOIN PAYROLL pa on pa.E_ID = e.E_ID
WHERE BASE_PAY = (SELECT MAX(BASE_PAY) FROM PAYROLL);
```

Explanation:

- Used MAX() to check the who gets highest base pay from all the Employees.

Output:

```
1 SELECT e.E_FNAME, e.E_MIDDLE_INIT, e.E_LNAME, pa.BASE_PAY
2 FROM employee e
3 JOIN PAYROLL pa ON pa.E_ID = e.E_ID
4 WHERE BASE_PAY = (SELECT max(BASE_PAY) FROM PAYROLL);
```

E_FNAME	E_MIDDLE_INIT	E_LNAME	BASE_PAY
AMRITH	K	AAKRE	24473.38

1 rows returned in 0.01 seconds [Download](#)

20. Give all the Office IDs which work under the Bureau of Health.

```
SELECT OFFICE_ID from OFFICE o
JOIN BUREAU b ON b.BUREAU_ID = o.BUREAU_ID
WHERE b.BUREAU_NAME = 'Bureau of Health';
```

Explanation:

- Here we filter all the office IDs for a particular Bureau.

Output:

```

1  SELECT OFFICE_ID FROM OFFICE o
2  JOIN BUREAU b ON b.BUREAU_ID = o.BUREAU_ID
3  WHERE b.BUREAU_NAME = 'Bureau of Health';

```

OFFICE_ID
4895
4897

21. Find the count of employees working in the 'DEPARTMENT OF CORRECTIONS' office.

Select COUNT(E_ID) as Number_of_Employees from EMP_OFFICE eo
 JOIN OFFICE o ON eo.OFFICE_ID = o.OFFICE_ID
 where OFFICE_NAME='DEPARTMENT OF CORRECTIONS';

Explanation:

- Used COUNT() to show the no. of employee works in the 'DEPARTMENT OF CORRECTIONS' office.

Output:

```

Select COUNT(E_ID) as Number_of_Employees from EMP_OFFICE eo
JOIN OFFICE o ON eo.OFFICE_ID = o.OFFICE_ID
where OFFICE_NAME='DEPARTMENT OF CORRECTIONS';

```

NUMBER_OF_EMPLOYEES
2

1 rows returned in 0.01 seconds [Download](#)

22. Display employees' names, hire dates, and office IDs.

SELECT E_FNAME, E_LNAME, E_HIRE_DATE, OFFICE_ID
 FROM EMPLOYEE e
 LEFT OUTER JOIN EMP_OFFICE eo ON e.E_ID = eo.E_ID;

Explanation:

- Showed list of all the employee names and their Hire dates using left Join to join EMP_OFFICE table.

Output:

```

1  SELECT E_FNAME, E_LNAME, E_HIRE_DATE, OFFICE_ID
2  FROM EMPLOYEE e
3  LEFT OUTER JOIN EMP_OFFICE eo ON e.E_ID = eo.E_ID;

```

E_FNAME	E_LNAME	E_HIRE_DATE	OFFICE_ID
AMRITH	AAKRE	06/16/2005	1250
TRES'SA	REED-HUMPHRIES	04/28/2008	1335
RUTH	ABARCA	06/03/2016	1032
TRACY	WILLIAMS	08/13/2001	4897
TRACY	ROBINSON	09/27/1999	1280
ZULEYMA	RIVERA-FARIAS	08/31/1998	4897
ZULEYMA	RIVERA-FARIAS	08/31/1998	1240
KHALIL	ABBOUD	12/17/1982	1280
ABDALLAH	ABDELHAMID	06/27/2011	1239
PAMELA	HUMPHRIES	11/18/2002	1240
PAMELA	WILLIAMSON	04/17/2006	1239
PAMELA	HUNT	11/01/1988	1326

12 rows returned in 0.05 seconds [Download](#)

23. Sum of base pay in Fiscal Year 2017.

```

SELECT SUM (BASE_PAY) AS Total_Base_Pay
FROM PAYROLL pa
JOIN FISCAL_YEAR f ON pa.FISCAL_PERIOD = f.FISCAL_PERIOD
Where FISCAL_YEAR = 2017;

```

Explanation:

- we are using Sum() to calculate the total sum of Base pay in 2017.

Output:

```

1  SELECT SUM (BASE_PAY) AS Total_Base_Pay
2  FROM PAYROLL pa
3  JOIN FISCAL_YEAR f
4  ON pa.FISCAL_PERIOD = f.FISCAL_PERIOD
5  WHERE FISCAL_YEAR = 2017;

```

TOTAL_BASE_PAY
360035.08

1 rows returned in 0.01 seconds [Download](#)

24. Display the name and job title of the employee having Employee ID '5'

```

SELECT DISTINCT e.E_FNAME, e.E_LNAME, e.E_ID, j.JOB_TITLE
FROM EMPLOYEE e JOIN PAYROLL pa ON E.E_ID = pa.E_ID
JOIN POSITION po ON po.POSITION_ID = pa.POSITION_ID
JOIN JOB j on j.JOB_CODE = po.JOB_CODE
WHERE e.E_ID= 5;

```

Explanation:

- Here we have used DISTINCT to show all the unique data for Employee id: 5.

Output:

```

1 SELECT DISTINCT e.E_FNAME, e.E_LNAME, e.E_ID, j.JOB_TITLE
2 FROM EMPLOYEE e JOIN PAYROLL pa ON E.E_ID = pa.E_ID
3 JOIN POSITION po ON po.POSITION_ID = pa.POSITION_ID
4 JOIN JOB j on j.JOB_CODE = po.JOB_CODE
5 WHERE e.E_ID= 5;
6

```

E_FNAME	E_LNAME	E_ID	JOB_TITLE
TRACY	ROBINSON	5	Supervisor Adult Probation

1 rows returned in 0.01 seconds [Download](#)

25. Find the count of the number of years from the date of hiring an employee has served. Display the employee's first and last name as full name.

```

SELECT E_ID, CONCAT(CONCAT(E_FNAME, ' '),
CONCAT(CONCAT(E_MIDDLE_INIT, ''), E_LNAME)) AS E_FULL_NAME,
TRUNC(TRUNC(MONTHS_BETWEEN(SYSDATE, E_HIRE_DATE)) / 12) AS
YEARS_OF_EXPERIENCE
FROM EMPLOYEE;

```

Explanation:

- Here we are displaying the employee's full name using the concat function and finding out the number of years the employee has been working in the company from the date of hire.

Output:

```

1 SELECT E_ID, CONCAT(CONCAT(E_FNAME, ' '), CONCAT(CONCAT(E_MIDDLE_INIT, ''), E_LNAME)) AS E_FULL_NAME,
2 TRUNC(TRUNC(MONTHS_BETWEEN(SYSDATE, E_HIRE_DATE)) / 12) AS YEARS_OF_EXPERIENCE
3 FROM EMPLOYEE;
4

```

E_ID	E_FULL_NAME	YEARS_OF_EXPERIENCE
1	AMRITH KAAKRE	17
2	TRESSA CREED-HUMPHRIES	15
3	RUTH AABARCA	6
4	TRACY WILLIAMS	21
5	TRACY RROBINSON	23
6	ZULEYMA VRIVERA-FARIAS	24
7	KHALIL EASBOUD	40
8	ABDALLAH FABELHAMID	11
9	PAMELA CHUMPHRIES	20
10	PAMELA R WILLIAMSON	17
11	PAMELA JHUNT	34

11 rows returned in 0.01 seconds [Download](#)

26. Find the Job title of the Employee with First name 'ZULEYMA' and Last name 'RIVERA-FARIAS'

```
Select JOB_TITLE, E_FNAME, E_LNAME from JOB j
JOIN POSITION p on j.JOB_CODE = p.JOB_CODE
JOIN PAYROLL py ON py.POSITION_ID = p.POSITION_ID
JOIN EMPLOYEE e ON e.E_ID = py.E_ID
WHERE e.E_Fname= 'ZULEYMA' AND e.E_Lname ='RIVERA-FARIAS';
```

Explanation:

- Here we are finding the job title of the Employee 'ZULEYMA RIVERA-FARIAS'. We have joined the POSITION, PAYROLL and EMPLOYEE tables to search the title.

Output:

```
1  Select JOB_TITLE, E_FNAME, E_LNAME from JOB j
2  JOIN POSITION p on j.JOB_CODE = p.JOB_CODE
3  JOIN PAYROLL py ON py.POSITION_ID = p.POSITION_ID
4  JOIN EMPLOYEE e ON e.E_ID = py.E_ID
5  WHERE e.E_Fname= 'ZULEYMA' AND e.E_Lname ='RIVERA-FARIAS';
```

Results	Explain	Describe	Saved SQL	History
JOB_TITLE		E_FNAME	E_LNAME	
Clerk V		ZULEYMA	RIVERA-FARIAS	
Clerk V		ZULEYMA	RIVERA-FARIAS	
Clerk V		ZULEYMA	RIVERA-FARIAS	
Clerk V		ZULEYMA	RIVERA-FARIAS	
Senior Clerk		ZULEYMA	RIVERA-FARIAS	
Senior Clerk		ZULEYMA	RIVERA-FARIAS	
Pharmacy Technician ARNTE OFH		ZULEYMA	RIVERA-FARIAS	
Pharmacy Technician ARNTE OFH		ZULEYMA	RIVERA-FARIAS	
Pharmacy Technician ARNTE OFH		ZULEYMA	RIVERA-FARIAS	
Pharmacy Technician ARNTE OFH		ZULEYMA	RIVERA-FARIAS	
Pharmacy Technician ARNTE OFH		ZULEYMA	RIVERA-FARIAS	
Pharmacy Technician ARNTE OFH		ZULEYMA	RIVERA-FARIAS	

27. Name of all Bureaus in Descending order

```
Select * from Bureau order by BUREAU_NAME DESC;
```

Explanation:

- Arranging all the Bureau names in Descending order.

Output:

```
Select * from Bureau order by BUREAU_NAME DESC;
```

Results Explain Describe Saved SQL History	
BUREAU_ID	BUREAU_NAME
5	State's Attorney
3	Sheriff
4	HEALTH
2	County Clerk
7	Clerk of Circuit Ct.
6	Chief Judge
8	CORPORATE
1	Bureau of Human Resources
9	Bureau of Health

9 rows returned in 0.38 seconds [Download](#)

28.What is the total number of employees hired in each fiscal year and each bureau?

```
SELECT fy.FISCAL_YEAR, b.BUREAU_NAME, COUNT(e.E_ID) AS
EMP_COUNT
FROM FISCAL_YEAR fy
JOIN PAYROLL p ON SUBSTR(p.FISCAL_PERIOD,1,6) = fy.FISCAL_PERIOD
JOIN EMPLOYEE e ON p.E_ID = e.E_ID
JOIN EMP_OFFICE h ON e.E_ID = h.E_ID
JOIN OFFICE o ON h.OFFICE_ID = o.OFFICE_ID
JOIN BUREAU b ON o.BUREAU_ID = b.BUREAU_ID
GROUP BY fy.FISCAL_YEAR, b.BUREAU_NAME
ORDER BY fy.FISCAL_YEAR, b.BUREAU_NAME;
```

Explanation:

- This SQL query retrieves the employee count per fiscal year and per bureau name. It joins several tables, including the FISCAL_YEAR, PAYROLL, EMPLOYEE, EMP_OFFICE, OFFICE, and BUREAU tables. The query calculates the employee count by counting the number of E_ID values in the EMPLOYEE table.

Output:

```
1  SELECT fy.FISCAL_YEAR, b.BUREAU_NAME, COUNT(e.E_ID) AS EMP_COUNT
2  FROM FISCAL_YEAR fy
3  JOIN PAYROLL p ON SUBSTR(p.FISCAL_PERIOD,1,6) = fy.FISCAL_PERIOD
4  JOIN EMPLOYEE e ON p.E_ID = e.E_ID
5  JOIN EMP_OFFICE h ON e.E_ID = h.E_ID
6  JOIN OFFICE o ON h.OFFICE_ID = o.OFFICE_ID
7  JOIN BUREAU b ON o.BUREAU_ID = b.BUREAU_ID
8  GROUP BY fy.FISCAL_YEAR, b.BUREAU_NAME
9  ORDER BY fy.FISCAL_YEAR, b.BUREAU_NAME;
```

FISCAL_YEAR	BUREAU_NAME	EMP_COUNT
2016	Bureau of Health	6
2016	Bureau of Human Resources	1
2016	Chief Judge	8
2016	Clerk of Circuit Ct.	4
2016	HEALTH	10
2016	Sheriff	8
2016	State's Attorney	4
2017	Bureau of Health	4
2017	Chief Judge	4
2017	Clerk of Circuit Ct.	4

29. Which employees were hired after January 9, 1998 and have the middle initial 'V'?

```
SELECT *
FROM EMPLOYEE
WHERE E_HIRE_DATE > 01-09-1998 AND E_MIDDLE_INIT = 'V';
```

Explanation:

- This SQL query selects all columns from the EMPLOYEE table where the E_HIRE_DATE column is greater than '01-09-1998' and the E_MIDDLE_INIT column equals 'V'. The query essentially returns all employees who were hired after January 9th, 1998 and have a middle initial of 'V'.

Output:

1	SELECT *
2	FROM EMPLOYEE
3	WHERE E_HIRE_DATE > '01-09-1998' AND E_MIDDLE_INIT = 'V';

Results	Explain	Describe	Saved SQL	History
E_ID	E_FNAME	E_LNAME	E_MIDDLE_INIT	E_HIRE_DATE
6	ZULEYMA	RIVERA-FARIAS	V	08/31/1998

30. Write a query to select the first and last name of employees who have been with the company for at least 10 years, sorted by hire date in ascending order.

```
SELECT E_FNAME, E_LNAME, E_HIRE_DATE
FROM EMPLOYEE
WHERE MONTHS BETWEEN (SYSDATE, E_HIRE_DATE) >= 120
ORDER BY E_HIRE_DATE ASC;
```

Explanation:

- This SQL query selects the first name, last name, and hires date of employees who have been employed for at least 120 months (i.e., 10 years) and sorts the result by hire date in ascending order. It uses the MONTHS_BETWEEN function to calculate the number of

months between the current date (SYSDATE) and the hire date, and filters the results to only include those where the result is greater than or equal to 120.

Output:

```
1  SELECT E_FNAME, E_LNAME, E_HIRE_DATE|
2  FROM EMPLOYEE
3  WHERE MONTHS_BETWEEN(SYSDATE, E_HIRE_DATE) >= 120
4  ORDER BY E_HIRE_DATE ASC;
5
```

KHALIL	ABBOUD	12/17/1982
PAMELA	HUNT	11/01/1988
ZULEYMA	RIVERA-FARIAS	08/31/1998
TRACY	ROBINSON	09/27/1999
TRACY	WILLIAMS	08/15/2001
PAMELA	HUMPHRIES	11/18/2002
AMRITH	AAKRE	07/16/2005
PAMELA	WILLIAMSON	04/17/2006
TRESSA	REED-HUMPHRIES	04/28/2008
ABDALLAH	ABDELHAMID	07/27/2011

31. Write a query to select the bureau name and the number of offices in that bureau, sorted by the number of offices in descending order.

```
SELECT b.BUREAU_NAME, COUNT(*) AS num_offices
FROM BUREAU b
JOIN OFFICE o ON b.BUREAU_ID = o.BUREAU_ID
GROUP BY b.BUREAU_NAME
ORDER BY num_offices DESC;
```

Explanation:

- This SQL query is retrieving information about the number of offices in each bureau and sorting the results in descending order based on the number of offices.

Output:

```
1 SELECT b.BUREAU_NAME, COUNT(*) AS num_offices
2 FROM BUREAU b
3 JOIN OFFICE o ON b.BUREAU_ID = o.BUREAU_ID
4 GROUP BY b.BUREAU_NAME
5 ORDER BY num_offices DESC;
```

BUREAU_NAME	NUM_OFFICES
Chief Judge	2
State's Attorney	1
Clerk of Circuit Ct.	1
Sheriff	1
CORPORATE	1
Bureau of Human Resources	1
County Clerk	1
Bureau of Health	1
HEALTH	1

32. Get the list of employees who have been hired before 2010, and whose job title is either 'Clerk V' or 'Senior Clerk'.

```
SELECT e.E_FNAME, e.E_LNAME, j.JOB_TITLE, e.E_HIRE_DATE
FROM EMPLOYEE e
JOIN PAYROLL p ON e.E_ID = p.E_ID
JOIN Position h ON p.POSITION_ID = h.POSITION_ID
JOIN Job j ON h.JOB_CODE = j.JOB_CODE
WHERE j.JOB_TITLE IN ('Clerk V', 'Senior Clerk')
```

AND e.E_HIRE_DATE < '01-01-2010';

Explanation:

- This SQL query retrieves the employee's first name, last name, job title, and hire date for employees who were hired before 2010 and whose job titles are either "Clerk V" or "Senior Clerk". The query joins four tables: EMPLOYEE, PAYROLL, Position, and Job.

Output:

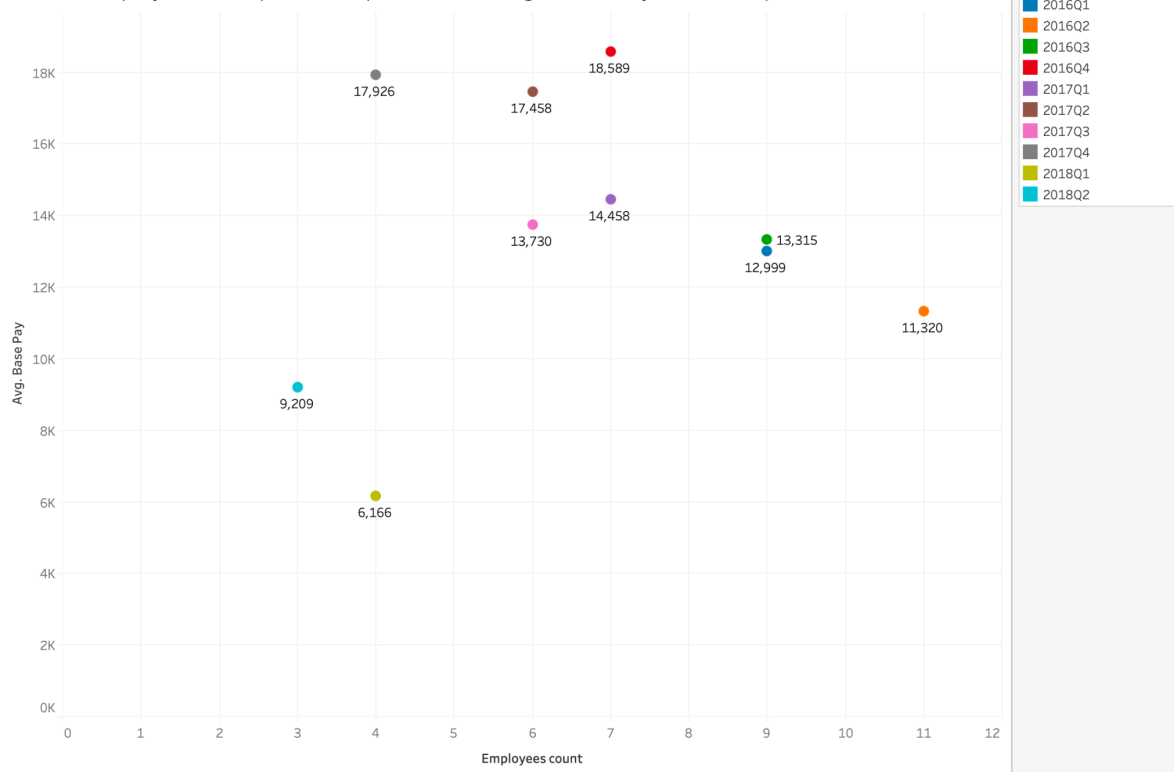
```
1 SELECT e.E_FNAME, e.E_LNAME, j.JOB_TITLE, e.E_HIRE_DATE
2 FROM EMPLOYEE e
3 JOIN PAYROLL p ON e.E_ID = p.E_ID
4 JOIN Position h ON p.POSITION_ID = h.POSITION_ID
5 JOIN Job j ON h.JOB_CODE = j.JOB_CODE
6 WHERE j.JOB_TITLE IN ('Clerk V', 'Senior Clerk')
7 AND e.E_HIRE_DATE < '01-01-2010';
8
9
```

E_FNAME	E_LNAME	JOB_TITLE	E_HIRE_DATE
ZULEYMA	RIVERA-FARIAS	Senior Clerk	08/31/1998
ZULEYMA	RIVERA-FARIAS	Senior Clerk	08/31/1998
ZULEYMA	RIVERA-FARIAS	Clerk V	08/31/1998
ZULEYMA	RIVERA-FARIAS	Clerk V	08/31/1998
ZULEYMA	RIVERA-FARIAS	Clerk V	08/31/1998
ZULEYMA	RIVERA-FARIAS	Clerk V	08/31/1998

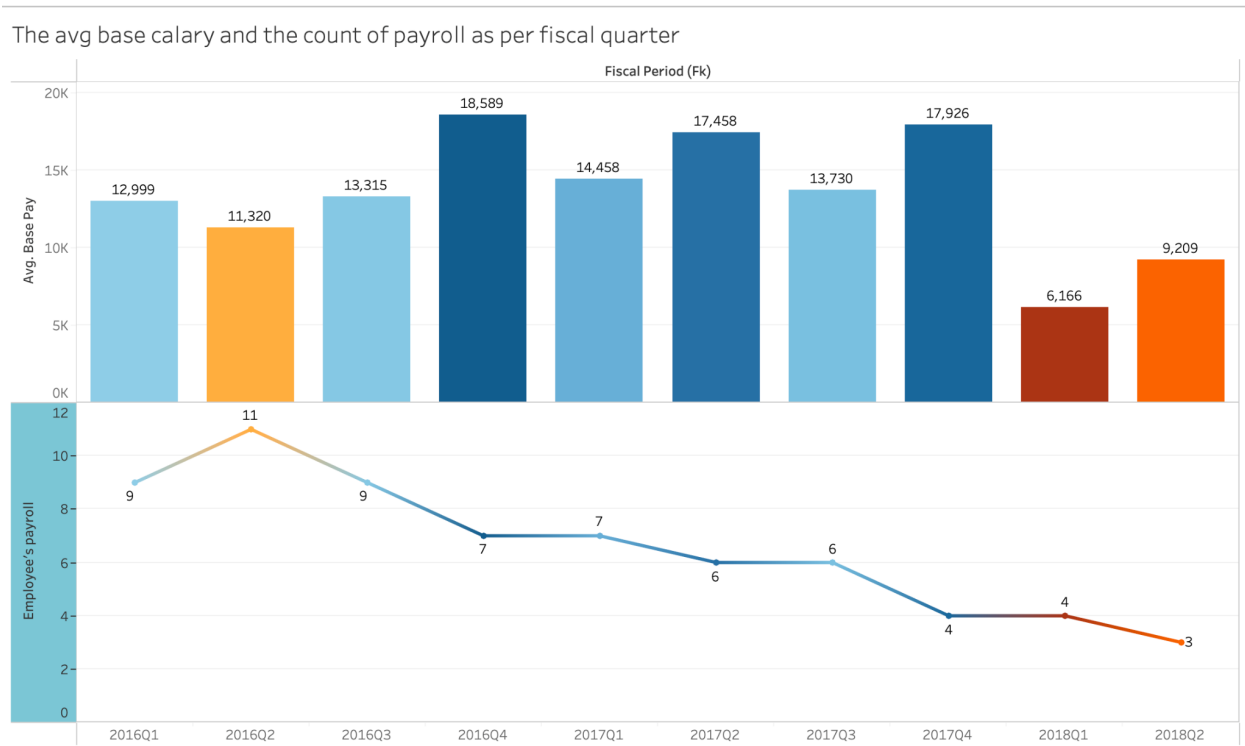
Rows returned in 0.03 seconds. [Download](#)

Visualizations

Count of Employee id's as per Fiscal quarters with avg base salary across the positions

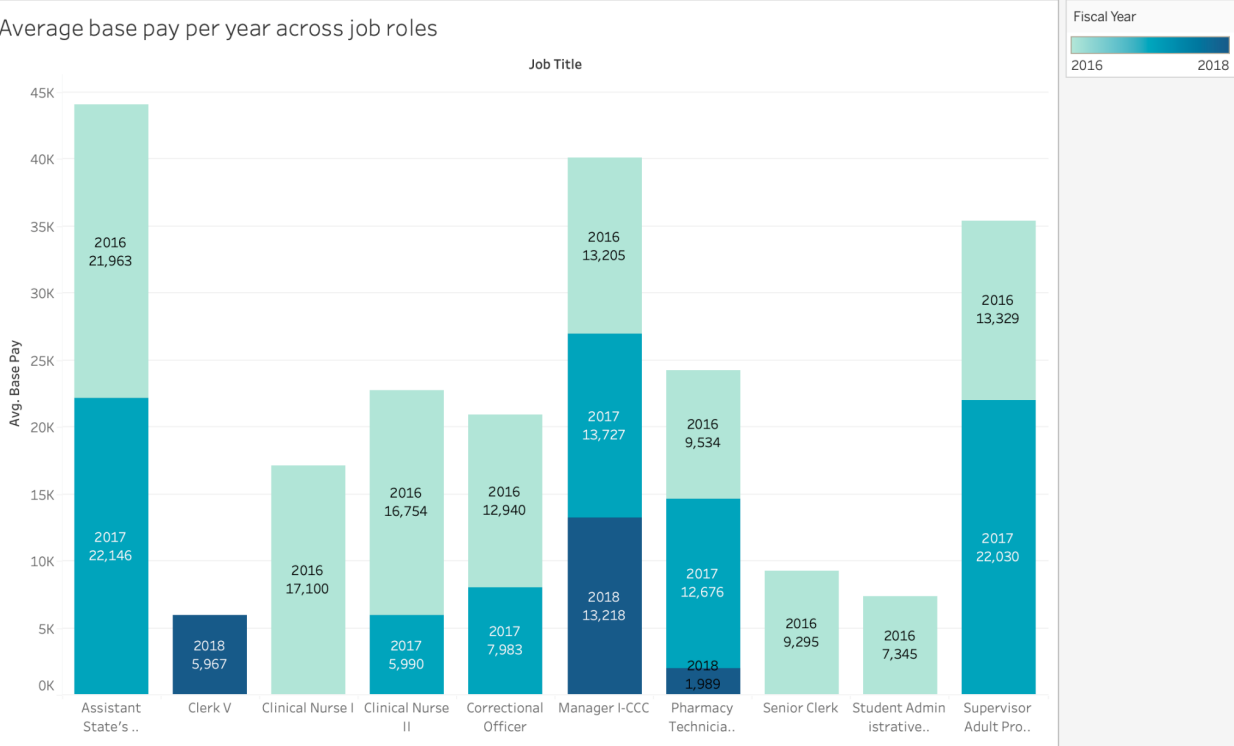


- The scatter diagram above shows how the number of employees and the average base salary for the several fiscal quarters relate to one another. We may see any patterns or trends that develop over time between these variables by graphically depicting the data points. Employee count is shown on the x-axis, and average base salary is shown on the y-axis. We can comprehend how variations in average base pay relate to changes in staff count by analyzing the scatter plot.



- The dual visualization provided above offers a detailed breakdown of how the count of employees has influenced compensation across specific quarters. The visualization mixes bar and line charts to offer a complete representation of the data. The line shows the matching personnel and the bars show the average base salary in relation to the fiscal quarters.
- By analyzing this visualization, we can gain insights into the relationship between employee count and compensation. It enables us to observe patterns and trends, helping us understand how changes in workforce size affect compensation within specific quarters.

Average base pay per year across job roles



- The final graphic above displays the association between job titles and their average base pay from 2016 to 2018. The compensation landscape is outlined in this image, which demonstrates how various job titles correspond to different average base wages throughout this three-year period.
- Analyzing this visualization, we can spot trends and patterns in remuneration across job titles. We can see which job titles have average base incomes that are more or lower than others, as well as how these salaries may have evolved.
- As a result, we may better comprehend the ranges in pay and choose occupations and compensation schemes by using the final visualization, which provides a glimpse of job titles and their typical base salary from 2016 to 2018.

Conclusion

To conclude, we have developed a database for managing information on employee payroll. That includes several tables such as JOB, POSITION, EMPLOYEE, PAYROLL, FISCAL_YEAR, OFFICE, BUREAU, and EMP_OFFICE. Additionally, we have designed an Entity Relationship Diagram that portrays the relationships between the entities. Finally, we have written queries and executed them in Oracle to display interaction with the database.