**SQL CODE EXAMPLES AND REAL-TIME SCENARIOS**

**To automate data extraction and reporting processes using SQL scripts.**

1. Display all employees hired between January 1, 2022, and December 31, 2022, and sort the result by the hire date.

SELECT employee\_id, first\_name, last\_name, hire\_date, salary

FROM employees

WHERE hire\_date BETWEEN '2022-01-01' AND '2022-12-31'

ORDER BY hire\_date;

* This SQL script will extract data of all employees who were hired between January 1, 2022, and December 31, 2022, and sort the result by the hire date.
* Further, we can save this script as a SQL file and execute it using a scheduling tool or command line interface to automate the data extraction process.

**We can save the result set in a CSV file using the following SQL script:**

SELECT employee\_id, first\_name, last\_name, hire\_date, salary

FROM employees

WHERE hire\_date BETWEEN '2022-01-01' AND '2022-12-31'

ORDER BY hire\_date

INTO OUTFILE '/path/to/file.csv'

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n';

* This SQL script will save the result set in a CSV file located at the specified file path.
* We can use the same scheduling tool or command line interface to automate the process of saving the result set in a CSV file.
* By automating these processes, we can generate HR reports more efficiently.

**To generate HR reports using the CSV file:**

1. Import the CSV file into a data visualization tool such as Tableau, Power BI, or Excel.
2. Choosing the appropriate visualization format, such as bar chart, line chart, or table, depending on the type of HR report you want to generate.
3. Selecting the data fields that we want to include in the HR report, such as employee ID, first name, last name, hire date, salary, and any other relevant fields.
4. Customize the report by adding filters, sorting options, and formatting to make the report easier to understand.
5. Save the report in a format that can be shared with others, such as PDF, PowerPoint, or Excel.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Calculate each employee's total compensation based on their hourly rate and hours worked?

SELECT employee\_id,

 (hourly\_rate \* hours\_worked + overtime\_pay) AS total\_compensation

FROM employee;

This SQL query extracts data from table called "employee" with columns for "employee\_id", "hourly\_rate", "hours\_worked", and "overtime\_pay". The query calculates the total compensation for each employee by multiplying their hourly rate by the number of hours worked, and adding any overtime pay. The results will display the employee ID and their total compensation.

**Explanation of how the calculation works:**

The expression "(hourly\_rate \* hours\_worked + overtime\_pay)" calculates the total compensation for each employee. This expression multiplies the employee's hourly rate by the number of hours worked, and adds any overtime pay.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***How to automate payroll process:**

1. Calculate employee compensation, taxes, and deductions based on data stored in a database. calculate employee gross pay, federal tax, state tax, and net pay?

We had to retrieve the employee ID, first name, last name, hourly rate, hours worked, gross pay, federal tax, state tax, and net pay.

SELECT

  employee\_id,

  first\_name,

  last\_name,

  hourly\_rate,

  hours\_worked,

  hourly\_rate \* hours\_worked AS gross\_pay,

  gross\_pay \* federal\_tax\_rate AS federal\_tax,

  gross\_pay \* state\_tax\_rate AS state\_tax,

  gross\_pay - federal\_tax - state\_tax AS net\_pay

FROM

  employee,

  tax\_rates

WHERE

  employee.state = tax\_rates.state;

**Query explanation:**

* The query extracts data from table called "employee" with columns for "employee\_id", "first\_name", "last\_name", "hourly\_rate", and "hours\_worked".
* It also extracts data from another table called "tax\_rates" with columns for "state", "federal\_tax\_rate", and "state\_tax\_rate".
* The query calculates each employee's gross pay by multiplying their hourly rate by the number of hours worked, and then calculates federal and state taxes and net pay based on the tax rates for the employee's state.

**Explanation of how the calculation works:**

* The expression "hourly\_rate \* hours\_worked" calculates the gross pay for each employee. This expression multiplies the employee's hourly rate by the number of hours worked.
* The expressions "gross\_pay \* federal\_tax\_rate" and "gross\_pay \* state\_tax\_rate" calculate the federal and state taxes for each employee, respectively. These expressions multiply the employee's gross pay by the federal and state tax rates for their state.
* The expression "gross\_pay - federal\_tax - state\_tax" calculates the net pay for each employee. This expression subtracts the federal and state taxes from the employee's gross pay.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Automate time and attendance tracking using sql**

1. How to track time and attendance use data from an HR system that tracks employee attendance and time off?

SELECT

  employee\_id,

  punch\_in,

  punch\_out,

  time\_off\_type

FROM

  attendance\_data

WHERE

  punch\_in BETWEEN '2023-01-01' AND '2023-01-31'

  AND time\_off\_type IS NULL;

* This SQL code selects time and attendance data from the "attendance\_data" table for the month of January 2023, but excludes any records for time off.
* The resulting data will include the employee ID, punch in time, punch out time, and any other relevant fields from the "attendance\_data" table.

Once we extract the time and attendance data, we can use SQL to generate reports and analytics.

For example, a query to calculate the total hours worked by each employee for the month of January 2023:

SELECT

  employee\_id,

  SUM(DATEDIFF(minute, punch\_in, punch\_out)) / 60.0 AS hours\_worked

FROM

  attendance\_data

WHERE

  punch\_in BETWEEN '2023-01-01' AND '2023-01-31'

  AND time\_off\_type IS NULL

GROUP BY

  employee\_id;

This query calculates the total hours worked by each employee for the month of January 2023, excluding any records for time off. The results will display the employee ID and total hours worked.

**Explanation of how the query works:**

* The SELECT statement specifies the columns to be retrieved from the "attendance\_data" table. In this case, we want to retrieve the employee ID and the total hours worked.
* The SUM function is used to add up the total number of minutes worked by each employee. The DATEDIFF function is used to calculate the difference between the punch in and punch out times in minutes.
* The / 60.0 expression is used to convert the total number of minutes worked to hours.
* The WHERE clause is used to filter the data based on certain conditions. In this case, we want to retrieve data for the month of January 2023, and exclude any records for time off.
* The GROUP BY clause groups the data by employee ID, so that the total hours worked can be calculated separately for each employee.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Some commonly used sql queries to generate reports based on the requirement.**

-- Headcount report by department and job title

SELECT Department, JobTitle, COUNT(\*) AS Headcount

FROM Employees

GROUP BY Department, JobTitle;

-- Turnover report by department

SELECT Department, COUNT(\*) AS TotalEmployees, SUM(CASE WHEN TerminationDate IS NOT NULL THEN 1 ELSE 0 END) AS Terminations,

(CAST(SUM(CASE WHEN TerminationDate IS NOT NULL THEN 1 ELSE 0 END) AS FLOAT) / COUNT(\*) \* 100) AS TurnoverPercentage

FROM Employees

GROUP BY Department;

-- Diversity report by department and race/ethnicity

SELECT Department, RaceEthnicity, COUNT(\*) AS Headcount

FROM Employees

GROUP BY Department, RaceEthnicity;

**Identify trends and patterns:**

-- Identify departments with high turnover rates

SELECT Department, COUNT(\*) AS TotalEmployees, SUM(CASE WHEN TerminationDate IS NOT NULL THEN 1 ELSE 0 END) AS Terminations,

(CAST(SUM(CASE WHEN TerminationDate IS NOT NULL THEN 1 ELSE 0 END) AS FLOAT) / COUNT(\*) \* 100) AS TurnoverPercentage

FROM Employees

WHERE HireDate >= '2019-01-01' AND TerminationDate IS NOT NULL

GROUP BY Department

HAVING TurnoverPercentage > 15;

-- Identify employees with high performance ratings

SELECT Name, JobTitle, PerformanceRating

FROM Employees

WHERE PerformanceRating >= 4;

**Conduct data analysis:**

-- Regression analysis to identify factors that impact employee turnover

SELECT JobTitle, Department, Salary, PerformanceRating, HireDate, TerminationDate

FROM Employees

WHERE HireDate >= '2019-01-01' AND TerminationDate IS NOT NULL;

-- Predictive modeling to forecast headcount by department

SELECT Department, COUNT(\*) AS Headcount, YEAR(HireDate) AS Year, MONTH(HireDate) AS Month

FROM Employees

GROUP BY Department, YEAR(HireDate), MONTH(HireDate);

**Troubleshoot data issues:**

-- Query to identify employees with missing data

SELECT \*

FROM Employees

WHERE Department IS NULL OR JobTitle IS NULL OR Salary IS NULL;

-- Query to identify discrepancies between HR and payroll systems

SELECT e.EmployeeID, e.Name, e.Department, e.JobTitle, e.Salary, p.Salary AS PayrollSalary

FROM Employees e

JOIN Payroll p ON e.EmployeeID = p.EmployeeID

WHERE e.Salary <> p.Salary;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**DATA MAPPING**

**data mapping for HR data migration:**

**Scenario 1**

**Migrating Employee Data:** Let's say an organization is migrating from an old HR system to a new HR system. The old system may have a field called "Employee ID," which is a unique identifier for each employee. The new system may have a similar field called "Employee Number." The data mapping process involves mapping the "Employee ID" field in the old system to the "Employee Number" field in the new system, ensuring that the employee data is accurately transferred.

**Scenario 2**

**Migrating Payroll Data:** During the migration of payroll data, the data mapping process involves mapping fields such as employee name, gross pay, net pay, deductions, and taxes from the old system to the new system. For example, the old system may have a field called "Federal Withholding Tax," and the new system may have a similar field called "Federal Taxes Withheld." The data mapping process involves mapping the old field to the new field, ensuring that the payroll data is accurately transferred.

**Scenario 3**

**Migrating Benefits Data:** During the migration of benefits data, the data mapping process involves mapping fields such as employee name, dependent information, plan type, coverage amount, and premium from the old system to the new system. For example, the old system may have a field called "Dental Plan Type," and the new system may have a similar field called "Dental Plan Coverage." The data mapping process involves mapping the old field to the new field, ensuring that the benefits data is accurately transferred.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**SECURITY**

Security configurations that we can implement to ensure the confidentiality, integrity, and availability of HR data in the HRIS system.

**Scenario-based examples on how to implement security configurations in an HRIS system.**

**Scenario 1**

An HR manager needs to access employee personal information such as Social Security numbers and home addresses for payroll processing, but other HR staff members don't need access to this information.

Solution: Set up user access controls to restrict access to employee personal information based on user roles and permissions. Create a user role for HR managers and assign them permissions to access employee personal information. Assign different roles to other HR staff members based on their job responsibilities and restrict their access to employee personal information.

**Scenario 2**

A recruiter logs into the HRIS system from a public Wi-Fi network and enters their password, but their account gets hacked, and the hacker gains access to confidential HR data.

Solution: Implement two-factor authentication to provide an extra layer of security for user logins. Require users to enter a one-time password that is sent to their mobile phone or email address in addition to their regular username and password. This will ensure that even if a user's password is compromised, the hacker won't be able to access the HRIS system without the one-time password.

**Scenario 3**

An employee accidentally deletes their own record from the HRIS system, and there is no way to recover the data.

Solution: Set up audit logs to track user activity in the HRIS system. Log user logins, logouts, failed login attempts, and changes to HR data. This will help you identify any unauthorized access or suspicious activity, and also help you recover deleted data if necessary.

**Scenario 4**

A hacker gains unauthorized access to the HRIS system and steals sensitive HR data such as employee Social Security numbers and salary information.

Solution: Encrypt sensitive HR data such as employee personal information and salary data to protect it from unauthorized access. Use encryption algorithms such as AES or RSA to encrypt data at rest and in transit. This will ensure that even if a hacker gains access to the HRIS system, they won't be able to read or steal the encrypted data.

**Scenario 5**

A virus infects the HRIS system and corrupts HR data, causing data loss and downtime.

Solution: Install firewalls and antivirus software to protect the HRIS system from external threats such as malware and viruses. Configure firewalls to restrict access to the HRIS system from unauthorized IP addresses, and install antivirus software to scan the system for any potential threats. This will help you prevent virus infections and other external threats from affecting the HRIS system and causing data loss or downtime.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**DATA VALIDATION**

**Data validation is an important aspect of ensuring the accuracy and completeness of HR data.**

**Scenario 1**

**New hire data entry:** When a new employee joins a company, there is a need to ensure that their data is entered accurately into the HR system. This includes their name, job title, start date, salary, benefits information, etc. Validate this data by comparing it against the information provided on the employee's offer letter, employment contract, and other relevant documents. If any discrepancies are found, they would need to be resolved before the data is saved in the HR system.

**Scenario 2**

**Employee data updates:** When an employee's data needs to be updated, such as a change in job title or salary, I would need to validate the updated information before saving it in the system. This would involve verifying the accuracy of the information provided by the employee or their manager and ensuring that it is in compliance with company policies and procedures.

**Scenario 3**

**Benefits enrollment:** During open enrollment periods, employees may make changes to their benefits selections. I would need to validate the accuracy of these changes by verifying that the employees’ selections match their benefits eligibility, and that the employee has provided all necessary information to enroll in the selected benefits. This would involve reviewing enrollment forms, benefit plan summaries, and other relevant documents to ensure that the employee's selections are valid.

**Scenario 4**

**Payroll data validation:** I would also need to validate payroll data to ensure that employee salaries, bonuses, and other compensation-related information are accurate. This would involve reviewing payroll reports and comparing the data against employment contracts, time and attendance records, and other relevant documents.