Document HR Analytics SQL queries

Step 1: Data Preparation

Data Import:

```
-- Create schema fOR HR data
CREATE SCHEMA IF NOT EXISTS hr_analytics;

-- Switch to the new schema
USE hr_analytics;

-- Create the hr_data table
CREATE TABLE hr_data (
 Age INT,
 AttritiON VARCHAR(3),
 BusinessTravel VARCHAR(50),
 DailyRate INT,
 Department VARCHAR(50),
 DistanceFromHome INT,
 EducatiON INT,
 EducationField VARCHAR(50),
 EmployeeCount INT,
 EmployeeNumber INT PRIMARY KEY,
 EnvironmentSatisfactiON INT,
 Gender VARCHAR(10),
 HourlyRate INT,
 JobInvolvement INT,
 JobLevel INT,
 JobRole VARCHAR(50),
 JobSatisfactiON INT,
 MaritalStatus VARCHAR(20),
 MonthlyIncome INT,
 MonthlyRate INT,
 NumCompaniesWorked INT,
 Over18 CHAR(1),
 OverTime VARCHAR(3),
 PercentSalaryHike INT,
 PerformanceRating INT,
 RelationshipSatisfactiON INT,
 StandardHours INT,
 StockOptionLevel INT,
 TotalWorkingYears INT,
 TrainingTimesLastYear INT,
 WorkLifeBalance INT,
```

```
 YearsAtCompany INT,
 YearsInCurrentRole INT,
 YearsSinceLastPromotiON INT,
 YearsWithCurrManager INT
);

-- Load data
    FROM CSV file
LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/hr_data.csv'
INTO TABLE hr_data
FIELDS TERMINATED BY ',
        '
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS
(Age,
        Attrition,
        BusinessTravel,
        DailyRate,
        Department,
        DistanceFromHome,
        Education,
        EducationField,
        EmployeeCount,
        EmployeeNumber,
        EnvironmentSatisfaction,
        Gender,
        HourlyRate,
        JobInvolvement,
        JobLevel,
        JobRole,
        JobSatisfaction,
        MaritalStatus,
        MonthlyIncome,
        MonthlyRate,
        NumCompaniesWorked,
        Over18,
        OverTime,
        PercentSalaryHike,
        PerformanceRating,
        RelationshipSatisfaction,
        StandardHours,
        StockOptionLevel,
        TotalWorkingYears,
        TrainingTimesLastYear,
        WorkLifeBalance,
        YearsAtCompany,
        YearsInCurrentRole,
```

```
        YearsSinceLastPromotion,
        YearsWithCurrManager);

-- Display first few rows to verify data
SELECT *
    FROM hr_data
LIMIT 5;
```

Output:

EmployeeID	FirstName	LastName	Gender	Age	BusinessTravel	Department	DistanceFromHome	State	Ethnicity	Education	EducationField	JobRole	MaritalStatus	Salary	StockOptionLevel	OverTime	HireDate	Attrition	YearsAtCompany	YearsInMostRecentRole	YearsSinceLastPromotion	YearsWithCurrManager	ReviewDate	EnvironmentSatisfaction	JobSatisfaction	RelationshipSatisfaction	TrainingOpportunitiesWithinYear	TrainingOpportunitiesTaken	WorkLifeBalanceRating	SelfRating	ManagerRating
3012-1A41	Leonelle	Simco	Female	30	Some Travel	Sales	27	IL	White	Doctorate	Marketing	Sales Executive	Divorced	102059	1	No	2012-01-03	No	10	4	9	7	2014-10-31	Neutral	Satisfied	Dissatisfied	1	0	Meets Expectation	4	3
3012-1A41	Leonelle	Simco	Female	30	Some Travel	Sales	27	IL	White	Doctorate	Marketing	Sales Executive	Divorced	102059	1	No	2012-01-03	No	10	4	9	7	2019-10-30	Very Dissatisfied	Neutral	Dissatisfied	3	1	Meets Expectation	5	4
3012-1A41	Leonelle	Simco	Female	30	Some Travel	Sales	27	IL	White	Doctorate	Marketing	Sales Executive	Divorced	102059	1	No	2012-01-03	No	10	4	9	7	2018-10-30	Very Satisfied	Very Satisfied	Satisfied	3	0	Exceeds Expectation	5	5
3012-1A41	Leonelle	Simco	Female	30	Some Travel	Sales	27	IL	White	Doctorate	Marketing	Sales Executive	Divorced	102059	1	No	2012-01-03	No	10	4	9	7	2017-10-30	Satisfied	Satisfied	Very Satisfied	3	1	Needs Improvement	3	2
3012-1A41	Leonelle	Simco	Female	30	Some Travel	Sales	27	IL	White	Doctorate	Marketing	Sales Executive	Divorced	102059	1	No	2012-01-03	No	10	4	9	7	2016-10-30	Neutral	Neutral	Dissatisfied	3	0	Exceeds Expectation	3	3

Data Cleaning:

```
-- Step 1: Remove duplicates based
    ON EmployeeID
        AND ReviewDate
-- Create a temporary table to store DISTINCT records
CREATE TEMPORARY TABLE temp_hr_data AS
SELECT DISTINCT *
FROM hr_data;

-- Delete ALL records
    FROM hr_data
DELETE
    FROM hr_data;

-- Insert the DISTINCT records back into hr_data
INSERT INTO hr_data
SELECT *
    FROM temp_hr_data;
```

```sql
-- Drop the temporary table
DROP TEMPORARY TABLE temp_hr_data;

-- Step 2: Standardize categorical values
--UPDATE Gender to ensure consistent values (e.g., male
        AND female should be correctly typed)
UPDATE hr_data
SET Gender = CASE

    WHEN Gender = 'male' THEN
    'Male'

    WHEN Gender = 'female' THEN
    'Female'

    ELSE Gender
END;

--UPDATE OverTime to ensure consistent values
UPDATE hr_data
SET OverTime = CASE

    WHEN OverTime = 'yes' THEN
    'Yes'

    WHEN OverTime = 'no' THEN
    'No'

    ELSE OverTime
END;

--UPDATE MaritalStatus to ensure consistent values
UPDATE hr_data
SET MaritalStatus = CASE

    WHEN MaritalStatus = 'single' THEN
    'Single'

    WHEN MaritalStatus = 'married' THEN
    'Married'

    WHEN MaritalStatus = 'divorced' THEN
    'Divorced'

    WHEN MaritalStatus = 'widowed' THEN
    'Widowed'

    ELSE MaritalStatus
END;
```

```sql
--UPDATE AttritiON to ensure consistent values
UPDATE hr_data
SET AttritiON = CASE

    WHEN AttritiON = 'yes' THEN
    'Yes'

    WHEN AttritiON = 'no' THEN
    'No'

    ELSE Attrition
END;

--UPDATE BusinessTravel to ensure consistent values
UPDATE hr_data
SET BusinessTravel = CASE

    WHEN BusinessTravel = 'some travel' THEN
    'Some Travel'

    WHEN BusinessTravel = 'travel frequently' THEN
    'Travel Frequently'

    WHEN BusinessTravel = 'non-travel' THEN
    'Non-Travel'

    ELSE BusinessTravel
END;

-- Step 3: Handle missing
        OR inconsistent values
-- SET default value fOR columns
WITH NULL values (example: filling Salary
WITH median)
UPDATE hr_data
SET Salary = 60000 -- Replace
WITH median
        OR another appropriate default value
WHERE Salary IS NULL;

-- Handle missing Gender by filling
WITH "Prefer NOT To Say" if NULL
UPDATE hr_data
SET Gender = 'Prefer NOT To Say'
WHERE Gender IS NULL;

-- Handle missing OverTime by filling
WITH 'No' if NULL
```

```sql
UPDATE hr_data
SET OverTime = 'No'
WHERE OverTime IS NULL;

-- Step 4: Removing records
WITH significant missing data (if needed)
-- Delete rows
    WHERE crucial columns (LIKE EmployeeID
        OR ReviewDate) are missing
DELETE
    FROM hr_data
WHERE EmployeeID IS NULL
        OR ReviewDate IS NULL;

-- Optional: Check for
        AND remove outliers
        OR impossible values
-- FOR example, if Age is less than 18
        OR greater than 100, you might want to remove those
DELETE
    FROM hr_data
WHERE Age < 18
        OR Age > 100;

-- Step 5: Final verificatiON of data consistency (optional)
-- Example: Verify that there are no missing values IN critical columns after
cleaning
SELECT COUNT(*) AS MissingValues
FROM hr_data
WHERE EmployeeID IS NULL
        OR ReviewDate IS NULL;

-- Step 6: Check fOR duplicates agaIN (ensure that no duplicate records
remain)
SELECT EmployeeID,
        ReviewDate,
        COUNT(*)
FROM hr_data
GROUP BY EmployeeID,
        ReviewDate
HAVING COUNT(*) > 1;


SELECT *
    FROM hr_data LIMIT 10;


SELECT EmployeeID,
        ReviewDate,
```

```
        COUNT(*)
FROM hr_data
GROUP BY EmployeeID,
        ReviewDate
HAVING COUNT(*) > 1;


SELECT COUNT(*) AS MissingValues
FROM hr_data
WHERE EmployeeID IS NULL
        OR ReviewDate IS NULL;


SELECT COUNT(*) AS total_rows
    FROM hr_analysis.dim_employee;
```

Output:

Data Normalization:

Code:

```
DELIMITER $$

-- Drop the procedure if it already exists
DROP PROCEDURE IF EXISTS create_hr_star_schema $$

-- Create the procedure again
CREATE PROCEDURE create_hr_star_schema()
BEGIN
 -- DimensiON Table: dim_employee
 CREATE TABLE IF NOT EXISTS hr_analysis.dim_employee (
 employee_id VARCHAR(50) PRIMARY KEY,
 first_name VARCHAR(50),
 last_name VARCHAR(50),
 gender ENUM('Male', 'Female', 'Non-Binary', 'Prefer NOT To Say'),
 age INT,
 business_travel VARCHAR(50),
 department VARCHAR(50),
 distance_from_home INT,
 state VARCHAR(50),
 ethnicity VARCHAR(50),
```

```
 job_role VARCHAR(100),
 marital_status ENUM('Single', 'Married', 'Divorced', 'Widowed'),
 salary INT,
 stock_option_level INT,
 over_time ENUM('Yes', 'No'),
 hire_date DATE,
 attritiON ENUM('Yes', 'No'),
 years_at_company INT,
 years_in_most_recent_role INT,
 years_since_last_promotiON INT,
 years_with_curr_manager INT
 );

 -- DimensiON Table: dim_educationlevel
 CREATE TABLE IF NOT EXISTS hr_analysis.dim_educationlevel (
 education_id INT AUTO_INCREMENT PRIMARY KEY,
 education_level VARCHAR(50) UNIQUE
 );

 -- DimensiON Table: dim_ratinglevel
 CREATE TABLE IF NOT EXISTS hr_analysis.dim_ratinglevel (
 rating_id INT AUTO_INCREMENT PRIMARY KEY,
 rating_level VARCHAR(50) UNIQUE
 );

 -- DimensiON Table: dim_satisfiedlevel
 CREATE TABLE IF NOT EXISTS hr_analysis.dim_satisfiedlevel (
 satisfaction_id INT AUTO_INCREMENT PRIMARY KEY,
 satisfaction_level VARCHAR(50) UNIQUE
 );

 -- Fact Table: fact_performancerating
 CREATE TABLE IF NOT EXISTS hr_analysis.fact_performancerating (
 fact_id INT AUTO_INCREMENT PRIMARY KEY,
 employee_id VARCHAR(50),
 review_date DATE,
 environment_satisfaction_id INT,
 job_satisfaction_id INT,
 relationship_satisfaction_id INT,
 training_opportunities_within_year INT,
 training_opportunities_taken INT,
 work_life_balance_rating_id INT,
 self_rating INT,
 manager_rating INT,
 FOREIGN KEY (employee_id) REFERENCES hr_analysis.dim_employee(employee_id),
 FOREIGN KEY (environment_satisfaction_id) REFERENCES
hr_analysis.dim_satisfiedlevel(satisfaction_id),
 FOREIGN KEY (job_satisfaction_id) REFERENCES
hr_analysis.dim_satisfiedlevel(satisfaction_id),
```

```
 FOREIGN KEY (relationship_satisfaction_id) REFERENCES
hr_analysis.dim_satisfiedlevel(satisfaction_id),
 FOREIGN KEY (work_life_balance_rating_id) REFERENCES
hr_analysis.dim_ratinglevel(rating_id)
 );

 -- Populate dimensiON tables
WITH DISTINCT values
    FROM hr_data

 -- Insert data into dim_employee
 INSERT IGNORE INTO hr_analysis.dim_employee (employee_id,
        first_name,
        last_name,
        gender,
        age,
        business_travel,
        department,
        distance_from_home,
        state,
        ethnicity,
        job_role,
        marital_status,
        salary,
        stock_option_level,
        over_time,
        hire_date,
        attrition,
        years_at_company,
        years_in_most_recent_role,
        years_since_last_promotion,
        years_with_curr_manager)
SELECT DISTINCT EmployeeID,
        FirstName,
        LastName,
        Gender,
        Age,
        BusinessTravel,
        Department,
        DistanceFromHome,
        State,
        Ethnicity,
        JobRole,
        MaritalStatus,
        Salary,
        StockOptionLevel,
        OverTime,
        HireDate,
        Attrition,
```

```
        YearsAtCompany,
        YearsInMostRecentRole,
        YearsSinceLastPromotion,
        YearsWithCurrManager

    FROM hr_data;

 -- Insert data into dim_educationlevel
 INSERT IGNORE INTO hr_analysis.dim_educationlevel (education_level)
SELECT DISTINCT Education

    FROM hr_data;

 -- Insert data into dim_ratinglevel
 INSERT IGNORE INTO hr_analysis.dim_ratinglevel (rating_level)
SELECT DISTINCT WorkLifeBalanceRating

    FROM hr_data;

 -- Insert data into dim_satisfiedlevel
 INSERT IGNORE INTO hr_analysis.dim_satisfiedlevel (satisfaction_level)
SELECT DISTINCT EnvironmentSatisfaction

    FROM hr_data
 UNION
SELECT DISTINCT JobSatisfaction

    FROM hr_data
 UNION
SELECT DISTINCT RelationshipSatisfaction

    FROM hr_data;

 -- Populate fact_performancerating table
 INSERT INTO hr_analysis.fact_performancerating (
 employee_id,
        review_date,
        environment_satisfaction_id,
        job_satisfaction_id,

 relationship_satisfaction_id,
        training_opportunities_within_year,
        training_opportunities_taken,

 work_life_balance_rating_id,
        self_rating,
        manager_rating
 )
SELECT
```

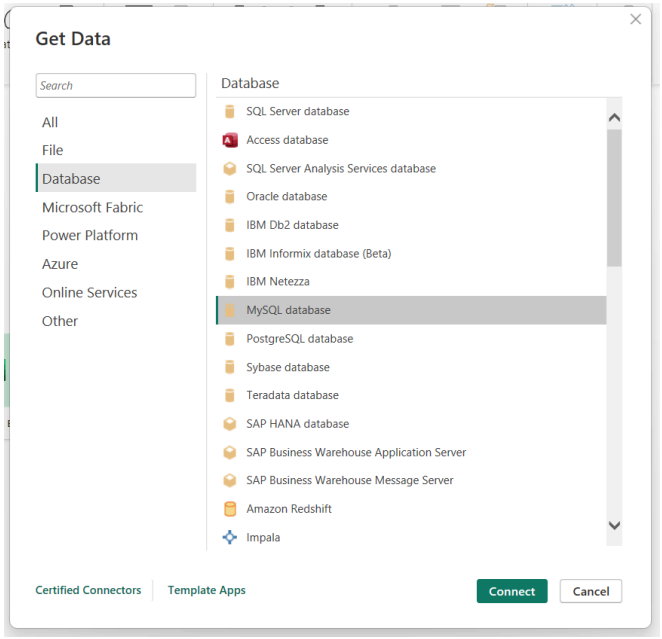
```
 e.employee_id,

 r.ReviewDate,

 es.satisfaction_id AS environment_satisfaction_id,

 js.satisfaction_id AS job_satisfaction_id,

 rs.satisfaction_id AS relationship_satisfaction_id,

 r.TrainingOpportunitiesWithinYear,

 r.TrainingOpportunitiesTaken,

 wl.rating_id AS work_life_balance_rating_id,

 r.SelfRating,

 r.ManagerRating

    FROM hr_data r

LEFT JOIN hr_analysis.dim_employee e
    ON r.EmployeeID = e.employee_id

LEFT JOIN hr_analysis.dim_satisfiedlevel es
    ON r.EnvironmentSatisfactiON = es.satisfaction_level

LEFT JOIN hr_analysis.dim_satisfiedlevel js
    ON r.JobSatisfactiON = js.satisfaction_level

LEFT JOIN hr_analysis.dim_satisfiedlevel rs
    ON r.RelationshipSatisfactiON = rs.satisfaction_level

LEFT JOIN hr_analysis.dim_ratinglevel wl
    ON r.WorkLifeBalanceRating = wl.rating_level;

END $$

DELIMITER ;

-- Call the procedure to create the HR star schema
CALL create_hr_star_schema();
```

Output:

job_role	marital_status	salary	stock_option_level	over_time	hire_date	attrition	years_a
Sales Executive	Married	56155	1	No	2017-08-26	No	5
Machine Learning Engineer	Married	126238	0	No	2012-03-08	No	10
Sales Executive	Married	97824	1	Yes	2020-03-16	Yes	1
Software Engineer	Single	68508	0	Yes	2012-01-28	Yes	5
Data Scientist	Single	109778	0	No	2022-06-23	Yes	0

fact_id	employee_id	review_date	environment_satisfaction_id	job_satisfaction_id	relationship_satisfaction_id	training_opportunities_within_year	training_opportunities_taken	work_life_balance_rating_id	self_
1	3012-1A41	2014-10-31	1	4	5	1	0	1	4
2	3012-1A41	2019-10-30	2	1	5	3	1	1	5
3	3012-1A41	2018-10-30	3	3	4	3	0	2	5
4	3012-1A41	2017-10-30	4	4	3	3	1	3	3
5	3012-1A41	2016-10-30	1	1	5	3	0	2	3

Step 2:


Get Data
Search
All
File
Database
Microsoft Fabric
Power Platform
Azure
Online Services
Other
Database
SQL Server database
Access database
SQL Server Analysis Services database
Oracle database
IBM Db2 database
IBM Informix database (Beta)
IBM Netezza
MySQL database
PostgreSQL database
Sybase database
Teradata database
SAP HANA database
SAP Business Warehouse Application Server
SAP Business Warehouse Message Server
Amazon Redshift
Impala
Certified Connectors
Template Apps
Connect
Cancel


MySQL database
Server
localhost
Database
hr_analysis
Advanced options
OK
Cancel