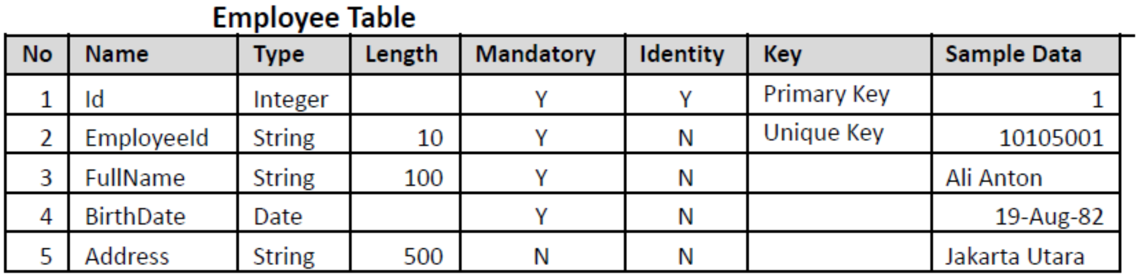
Akasia Answer for data engineer task

Ridwan Rahma Putra

Tables and Samples Data

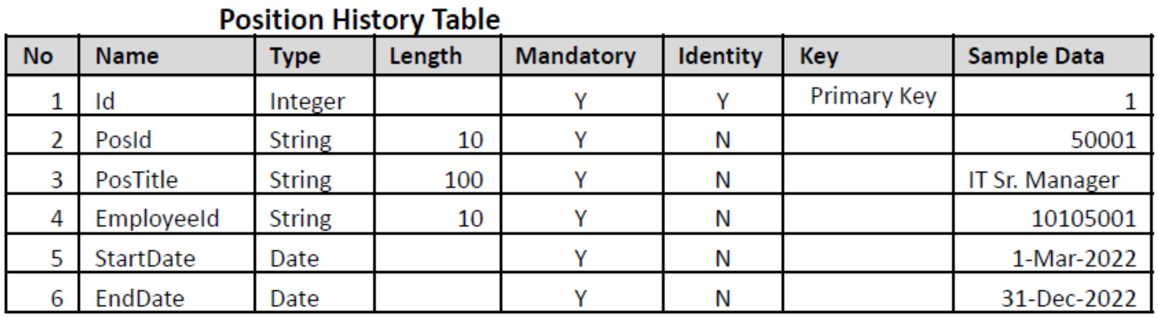
1. Employee Table



Employee Table

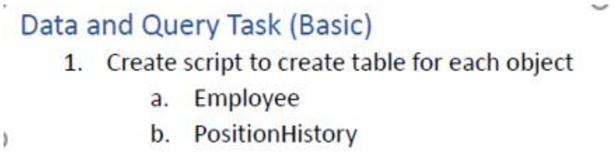
No	Name	Type	Length	Mandatory	Identity	Key	Sample Data
1	Id	Integer		Y	Y	Primary Key	1
2	EmployeeId	String	10	Y	N	Unique Key	10105001
3	FullName	String	100	Y	N		Ali Anton
4	BirthDate	Date		Y	N		19-Aug-82
5	Address	String	500	N	N		Jakarta Utara

2. Position Table



Position History Table

No	Name	Type	Length	Mandatory	Identity	Key	Sample Data
1	Id	Integer		Y	Y	Primary Key	1
2	PosId	String	10	Y	N		50001
3	PosTitle	String	100	Y	N		IT Sr. Manager
4	EmployeeId	String	10	Y	N		10105001
5	StartDate	Date		Y	N		1-Mar-2022
6	EndDate	Date		Y	N		31-Dec-2022



Data and Query Task (Basic)

1. Create script to create table for each object
   a. Employee
   b. PositionHistory

```
CREATE TABLE Employee (
    Id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    EmployeeId VARCHAR2(10) UNIQUE NOT NULL,
    FullName VARCHAR2(100) NOT NULL,
    BirthDate DATE NOT NULL,
    Address VARCHAR2(500)
);

CREATE TABLE PositionHistory (
    Id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    PosId VARCHAR2(10) NOT NULL,
    PosTitle VARCHAR2(100) NOT NULL,
    EmployeeId VARCHAR2(10) NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL,
    CONSTRAINT FK_Employee FOREIGN KEY (EmployeeId) REFERENCES Employee(EmployeeId)
);
```

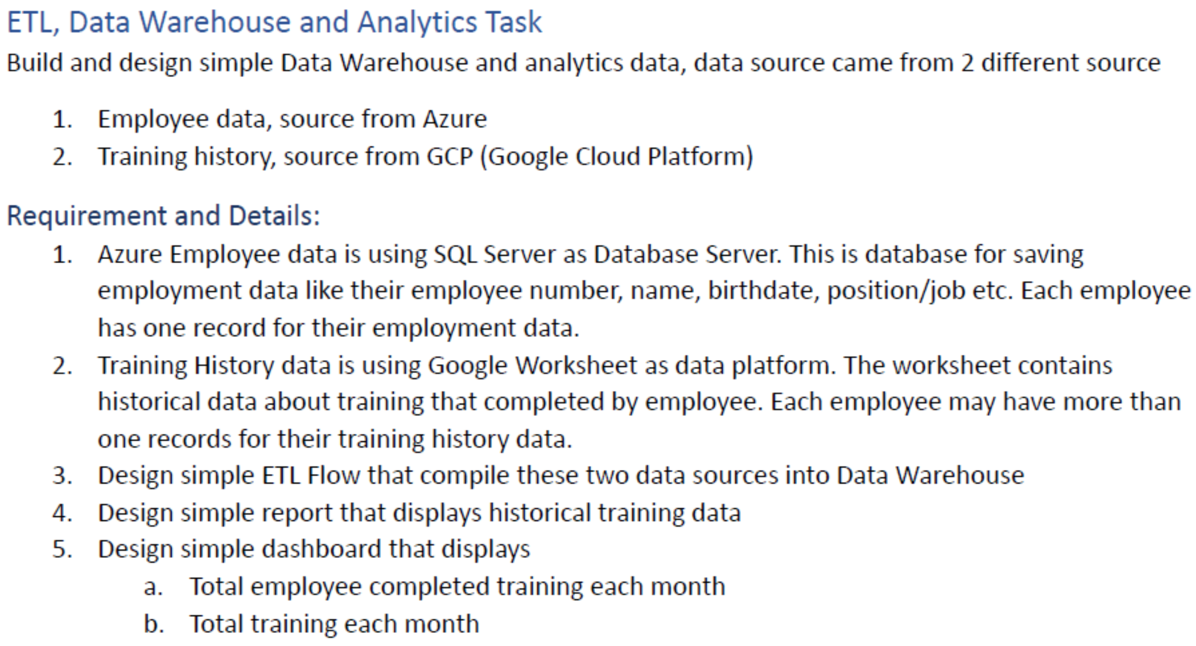
2. Create insert script to inserting data into each table (*Employee and PositionHistory*)

```
INSERT INTO Employee (EmployeeId, FullName, BirthDate, Address)
VALUES
('10105001', 'Ali Anton', TO_DATE('19-08-1982', 'DD-MM-YYYY'), 'Jakarta Utara');

INSERT INTO PositionHistory (PosId, PosTitle, EmployeeId, StartDate, EndDate)
VALUES
('50001', 'IT Sr. Manager', '10105001', TO_DATE('01-03-2022', 'DD-MM-YYYY'),
TO_DATE('31-12-2022', 'DD-MM-YYYY'));
```

3. Create query to display all employee (*EmployeeId, FullName, BirthDate, Address*) data with their **current** position information (*PosId, PosTitle, EmployeeId, StartDate, EndDate*).

```
select
  EmployeeId,
  FullName,
  BirthDate,
  Address,
  PosId,
  PosTitle,
  StartDate,
  EndDate
from
  (
    select
      e.EmployeeId,
      e.FullName,
      e.BirthDate,
      e.Address,
      ph.PosId,
      ph.PosTitle,
      ph.StartDate,
      ph.EndDate,
      row_number() over (partition by e.EmployeeId order by ph.StartDate desc) seq
    FROM
      Employee e LEFT JOIN PositionHistory ph ON e.EmployeeId = ph.EmployeeId
  ) a
where
  seq = 1
```



ETL, Data Warehouse and Analytics Task

Build and design simple Data Warehouse and analytics data, data source came from 2 different source

1. Employee data, source from Azure
2. Training history, source from GCP (Google Cloud Platform)

Requirement and Details:

1. Azure Employee data is using SQL Server as Database Server. This is database for saving employment data like their employee number, name, birthdate, position/job etc. Each employee has one record for their employment data.
2. Training History data is using Google Worksheet as data platform. The worksheet contains historical data about training that completed by employee. Each employee may have more than one records for their training history data.
3. Design simple ETL Flow that compile these two data sources into Data Warehouse
4. Design simple report that displays historical training data
5. Design simple dashboard that displays
   a. Total employee completed training each month
   b. Total training each month

Designing the Data Warehouse

The data warehouse will gather and maintain data for the purposes of reporting and analysis. We will create two fact tables along with their associated dimension tables:

1. Dim_Employee**:** Employee details (EmployeeID, Name, Birthdate, Position, etc.).
2. Fact_TrainingHistory**:** Training details (EmployeeID, TrainingName, CompletionDate, TrainingMonth, TrainingYear).

SQL Data Warehouse Table Design

Dim_Employee:

```
CREATE TABLE Dim_Employee (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100),
    Birthdate DATE,
    Position VARCHAR(50)
);
```

Fact_TrainingHistory:

```
CREATE TABLE Fact_TrainingHistory (
    TrainingID INT PRIMARY KEY,
    EmployeeID INT,
    TrainingName VARCHAR(100),
    CompletionDate DATE,
    TrainingMonth INT,
    TrainingYear INT,
    FOREIGN KEY (EmployeeID) REFERENCES Dim_Employee(EmployeeID)
);
```

Steps in the ETL Flow:

1. **Extract:**
    - Fetch data from Azure SQL Server for employee details via odbc.
    - Use Google Sheets API to fetch training history data from the Google Worksheet.
2. **Transform:**
    - Standardize data formats (for example dates format, ID).
    - Remove duplicates and clean data.
    - Create calculated columns like TrainingMonth and TrainingYear from CompletionDate.
3. **Load:**
    - Load Dim_Employee with employee data from Azure.
    - Load Fact_TrainingHistory with transformed training history data from Google Sheets.

Tools for ETL:

- Use an ELT tool like Apache NiFi / talend /pentaho
- If data small enough, use python

4. Designing the Reports

a. Historical Training Data Report

Query for generating historical training data:

```
SELECT
    e.EmployeeID,
    e.Name,
    e.Position,
    t.TrainingName,
    t.CompletionDate
FROM
    Fact_TrainingHistory t
JOIN
    Dim_Employee e ON t.EmployeeID = e.EmployeeID;
```

Dashboard Queries

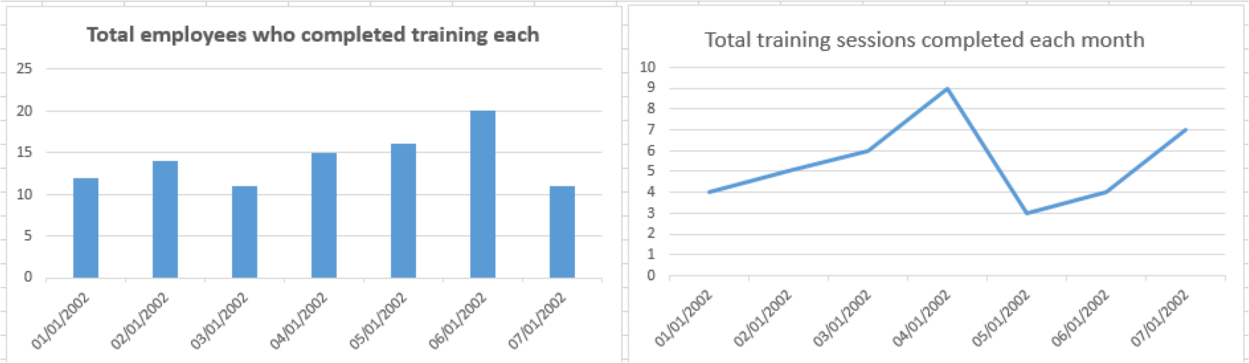
Total employees who completed training each month:

```
SELECT
    TrainingMonth,
    TrainingYear,
    COUNT(DISTINCT EmployeeID) AS TotalEmployees
FROM
    Fact_TrainingHistory
GROUP BY
    TrainingMonth, TrainingYear
ORDER BY
    TrainingYear, TrainingMonth;
```

Total training sessions completed each month:

```
SELECT
    TrainingMonth,
    TrainingYear,
    COUNT(TrainingID) AS TotalTrainings
FROM
    Fact_TrainingHistory
GROUP BY
    TrainingMonth, TrainingYear
ORDER BY
    TrainingYear, TrainingMonth;
```

Mockup chart


Total employees who completed training each
01/01/2002
02/01/2002
03/01/2002
04/01/2002
05/01/2002
06/01/2002
07/01/2002
Total training sessions completed each month
01/01/2002
02/01/2002
03/01/2002
04/01/2002
05/01/2002
06/01/2002
07/01/2002