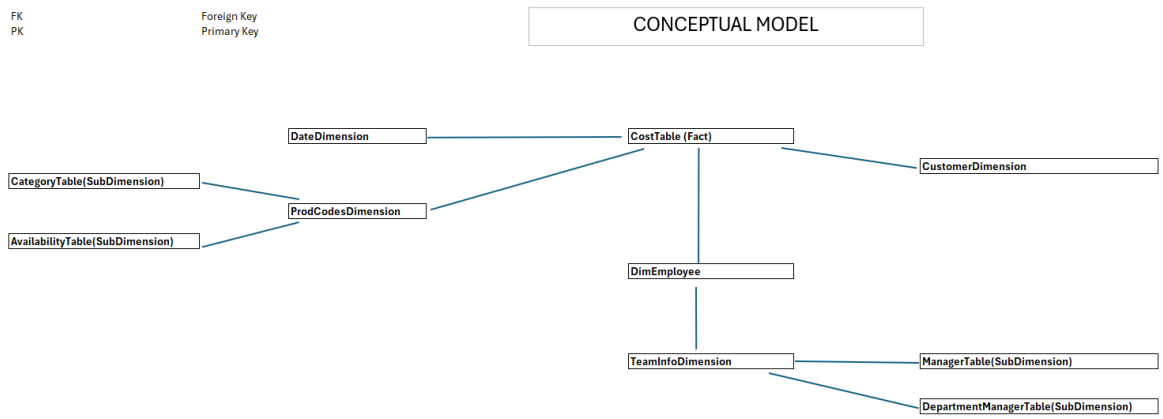
IGNITION - RESOURCE MANAGEMENT DATA MODEL AND DATABASE

The document below presents a Data Model for Resource sheet data, outlining the processes used to normalize, manage and store the data effectively.

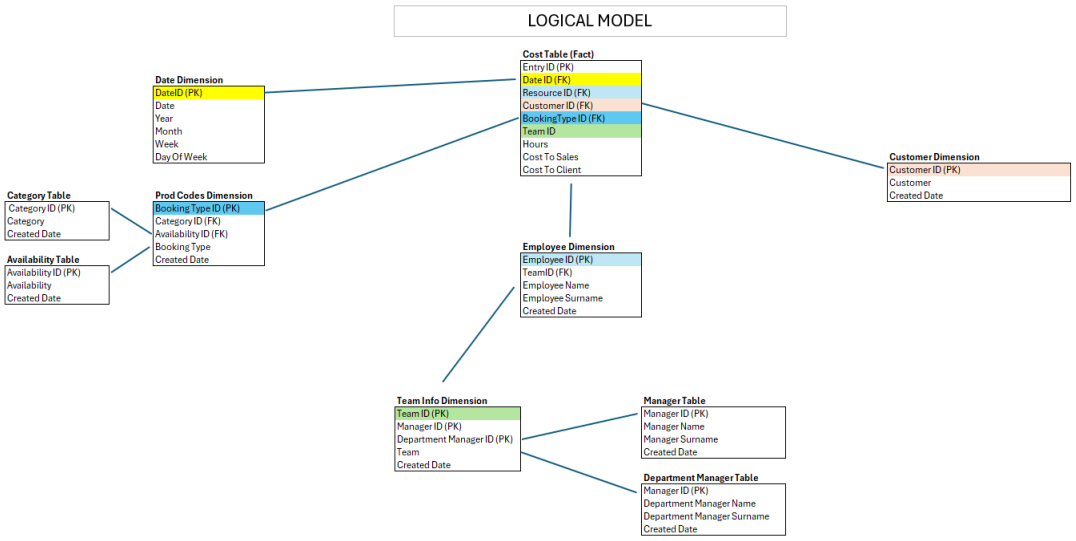
Step 1

I created a Conceptual Model first for High level representation of the data.



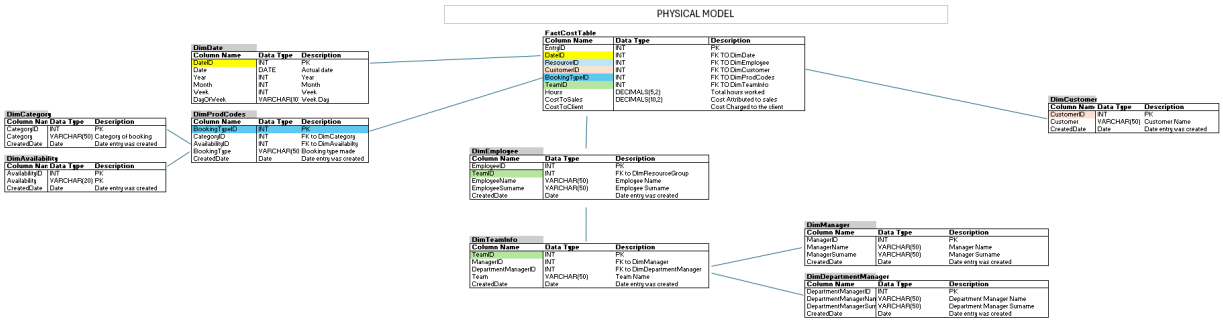
Step 2

I created a Logical Model to define the relationships between the attributes of each entity and the level of the relationships



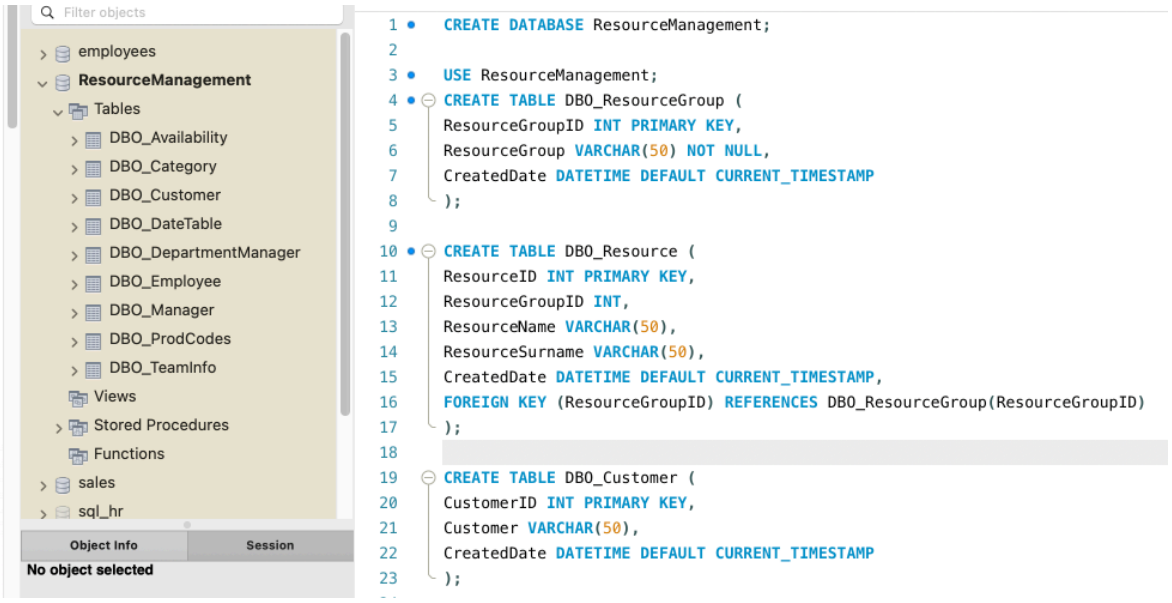
Step 3

I created a Physical Model to indicate how the data will be implemented into the database, also showing the attributes of each table to be used and the data types.



Step 4

I wrote in the code to be used to create the Database different tables (not all code is included to prevent the document from being too long).



Step 5

I created a stored procedure to populate the date table

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `PopulateDateTable`()
BEGIN
    DECLARE currentDate DATE;
    DECLARE endDate DATE;
    SET currentDate = '2020-01-01';
    SET endDate = '2030-12-31';

    WHILE currentDate <= endDate DO
        INSERT INTO DBO_DateTable (FullDate, Year, Month, MonthName, Week, DayOfWeek, IsWeekend)
        VALUES (
            currentDate,
            YEAR(currentDate),
            MONTH(currentDate),
            MONTHNAME(currentDate),
            WEEK(currentDate, 1),
            DAYNAME(currentDate),
            CASE WHEN DAYOFWEEK(currentDate) IN (1, 7) THEN TRUE ELSE FALSE END
        );

        SET currentDate = DATE_ADD(currentDate, INTERVAL 1 DAY);
    END WHILE;
END
```

Step 6

I Populated all the Dimension tables that will be used by the fact table

Example of some of the code used

```sql
USE ResourceManagement;
INSERT INTO DBO_Availability
(AvailabilityID, Availability)
VALUES
(1, 'Available'),
(2,'Unavailable');

# SELECT * FROM DBO_Availability

INSERT INTO DBO_Category
(CategoryID, Catergory)
VALUES
(1, 'Billable'),
(2, 'Productive'),
(3, 'Administrative'),
(4, 'Other');
```

```
109     ('Brendan Boyer','Project Nitrogen'),
110     ('Chaya Patterson','Project Nitrogen'),
111     ('Flora Ball','Manufacturing'),
112     ('Donovan Moran','Management'),
113     ('Persephone Meadows','Project Steam'),
114     ('Ronin Howe','Management'),
115     ('Amir Daniel','Project Nitrogen'),
116     ('Myles Larson','Wireless');
117
118 •   DROP TABLE TempEmployee;
119
120 •   INSERT INTO DBO_Employee (EmployeeID, TeamID, EmployeeName, EmployeeSurname )
121     SELECT EmployeeID,
122     CASE
123         WHEN te.Team = 'Management' THEN 40001
124         WHEN te.Team = 'Project Steam' THEN 40002
125         WHEN te.Team = 'Wireless' THEN 40003
126         WHEN te.Team = 'Infrastructure' THEN 40004
127         WHEN te.Team = 'Project Nitrogen' THEN 40005
128         WHEN te.Team = 'Manufacturing' THEN 40006
129         ELSE NULL
130         END AS TeamID,
131         SUBSTRING_INDEX(Employee, ' ', 1) AS EmployeeName,
132         SUBSTRING_INDEX(Employee, ' ',-1) AS EmployeeSurname
133     From TempEmployee te
```

Example of the tables

```sql
USE ResourceManagement;

SELECT * FROM DBO_Availability;

SELECT * FROM DBO_Customer;

SELECT * FROM DBO_Category;
```

100% 27:7

Result Grid Filter Rows: Search Edit:

AvailabilityID	Availability	CreatedDate
1	Available	2024-12-29 09:38:32
2	Unavailable	2024-12-29 09:38:32
NULL	NULL	NULL

DBO_Availability 8 | DBO_Customer 9 | DBO_Category 10

```sql
USE ResourceManagement;

SELECT * FROM DBO_Availability;

SELECT * FROM DBO_Customer;

SELECT * FROM DBO_Category;
```

100% 27:7

Result Grid Filter Rows: Search Edit:

CustomerID	Customer	CreatedDate
3	Hogbridge	2024-12-30 18:01:22
4	Wizardmaster	2024-12-30 18:01:22
5	Accentgate	2024-12-30 18:01:22
6	Timberhouse	2024-12-30 18:01:22
7	Herbbit	2024-12-30 18:01:22
8	Hammernite	2024-12-30 18:01:22
9	Kantouch	2024-12-30 18:01:22
10	Streettom	2024-12-30 18:01:22
11	Pinnaclecast	2024-12-30 18:01:22
12	Driftcast	2024-12-30 18:01:22
13	Lionman	2024-12-30 18:01:22
14	Heartex	2024-12-30 18:01:22
15	Spirit Star	2024-12-30 18:01:22
16	Volfase	2024-12-30 18:01:22
17	Joytechno	2024-12-30 18:01:22

DBO_Availability 8 | DBO_Customer 9 | DBO_Category 10

```sql
USE ResourceManagement;

SELECT * FROM DBO_Availability;

SELECT * FROM DBO_Customer;

SELECT * FROM DBO_Category;
```

100% 27:7

Result Grid Filter Rows: Search Edit:

CategoryID	Category	CreatedDate
1	Billable	2024-12-29 09:52:18
2	Productive	2024-12-29 09:52:18
3	Administrative	2024-12-29 09:52:18
4	Other	2024-12-29 09:52:18
NULL	NULL	NULL

DBO_Availability 8 | DBO_Customer 9 | DBO_Category 10

```sql
SELECT * FROM DBO_Employee
```

100% 27:9

Result Grid Filter Rows: Search Edit: Export/Import:

EmployeeID	TeamID	EmployeeName	EmployeeSurname	CreatedDate
5000	40002	Rohan	Ford	2024-12-31 14:37:57
5001	40002	Celeste	Hester	2024-12-31 14:37:57
5002	40001	Eliana	Jensen	2024-12-31 14:37:57
5003	40003	Dallas	Reid	2024-12-31 14:37:57
5004	40006	Shane	Lambert	2024-12-31 14:37:57
5005	40004	Nikolas	Orozco	2024-12-31 14:37:57
5006	40003	Lilyana	Melendez	2024-12-31 14:37:57
5007	40005	Rowan	Dougherty	2024-12-31 14:37:57
5008	40002	Valentin	Gibson	2024-12-31 14:37:57
5009	40004	Renata	Zhang	2024-12-31 14:37:57
5010	40001	Rowan	Holloway	2024-12-31 14:37:57
5011	40002	Katalina	Watkins	2024-12-31 14:37:57
5012	40002	Nash	Norman	2024-12-31 14:37:57
5013	40002	Alexandra	McMillan	2024-12-31 14:37:57
5014	40006	Aspen	Garner	2024-12-31 14:37:57

Step 7

I created a temporary table to ingest the original Resource Data as is before we can normalise it for easy and less memory usage storage.
Only 2 rows of data were used as this is for illustration purposes.

```sql
-- Create a temporary table to ingest the fact/ Resource data
CREATE TEMPORARY TABLE TempBookings (
    Date DATE,
    Hours DECIMAL(5, 2),
    Resource VARCHAR(100),
    ResourceGroup VARCHAR(100),
    Customer VARCHAR(100),
    BookingType VARCHAR(50),
    CosttoSales DECIMAL(10, 2),
    CosttoClient DECIMAL(10, 2)
);

-- Insert data into the temporary table
INSERT INTO TempBookings (Date, Hours, Resource, ResourceGroup, Customer, BookingType, CosttoSales, CosttoClient)
VALUES
('2023-02-24', 4.25, 'Eliana Jensen', 'Management', 'Sanit', 'PSBill-Client Funded', 156.00, 208.00),
('2023-01-31', 8.5, 'Dallas Reid', 'Wireless', 'Cloudtronics', 'PSBill-Client Funded', 581.00, 851.00);


SELECT * FROM TempBookings;
```

100% 57:19

Result Grid Filter Rows: Search Export:

Date	Hours	Resource	ResourceGroup	Customer	BookingType	CosttoSales	CosttoClie...
2023-02-24	4.25	Eliana Jensen	Management	Sanit	PSBill-Client Funded	156.00	208.00
2023-01-31	8.50	Dallas Reid	Wireless	Cloudtronics	PSBill-Client Funded	581.00	851.00

Step 8

I created a query/ Stored procedure to be used to populated the Resource Data (fact table) in a normalised manner in the database with the Resource data that will be provided from the excel sheet

```
DELIMITER $$

CREATE PROCEDURE InsertToCostTable()
BEGIN

INSERT INTO DBO_CostTable(
    DateID, ResourceID, CustomerID, BookingTypeID, TeamID, Hours, CostToSales, CostToClient
)
SELECT
    d.DateID,
     e.EmployeeID,
     c.CustomerID,
     b.BookingTypeID,
     t.TeamID,
     tb.Hours,
     tb.CosttoSales,
     tb.CosttoClient

FROM TempBookings tb
 INNER JOIN DBO_DateTable d ON d.FullDate = tb.Date
    INNER JOIN DBO_Employee e ON CONCAT(e.EmployeeName,' ',e.EmployeeSurname) = tb.Resource
    INNER JOIN DBO_Customer c ON c.Customer = tb.Customer
    INNER JOIN DBO_ProdCodes b ON b.BookingType = tb.BookingType
    LEFT JOIN DBO_TeamInfo t ON t.Team = tb.ResourceGroup;

END$$

```

PLEASE NOTE STEP 7 AND STEP 8 CAN BE WRITTEN INTO THE SAME PROCEDURE AND THE INGESTION OF THE RESOURCE DATA FILE CAN BE DONE VIA CODE TO AVOID READING IN THE WHOLE EXCEL DATA SET MANUALLY AND CREATE PROPER AUTOMATION OF THE DATA INGESTION

Once the fact table is recreated this is how the Table in the database will look like creating an easier to use and normalised, cost effective as it uses less space to store the data especially if and when keeping the historical data is important

```
13      b.BookingTypeID,
14      t.TeamID,
15      tb.Hours,
16      tb.CosttoSales,
17      tb.CosttoClient
18
19  FROM TempBookings tb
20   INNER JOIN DBO_DateTable d ON d.FullDate = tb.Date
21      INNER JOIN DBO_Employee e ON CONCAT(e.EmployeeName,' ',e.EmployeeSurname) = tb.Resource
22      INNER JOIN DBO_Customer c ON c.Customer = tb.Customer
23      INNER JOIN DBO_ProdCodes b ON b.BookingType = tb.BookingType
24      LEFT JOIN DBO_TeamInfo t ON t.Team = tb.ResourceGroup;
25
26  END$$
27  DELIMITER ;
28
29  SELECT * FROM DBO_CostTable
```

100% 28:29

Result Grid | Filter Rows: Search | Edit: | Export/Import:

ID	DateID	ResourceID	CustomerID	BookingTypeID	TeamID	Hours	CostToSales	CostToClient
1	1127	5003	2	4	4003	8.50	581.00	851.00
2	1151	5002	1	4	4001	4.25	156.00	208.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL