



ICBT
CAMPUS



Cardiff
Metropolitan
University

Database Design Development

STUDENT NAME: - P.C Pahalawanniarachchi

STUDENT ID: - CL/HDCSE/CMU/104/19

CARDIFF USER ID: - st20245800

Assignment Cover Sheet

Qualification		Module Number and Title
HD in Software Engineering/HD in Network Technology and Cybersecurity		CSE4005/Database Design Development
Student Name & No.		Assessor
Pasan Chamathka CL/HDCSE/C MU/104/19		MS.Vindya
Hand out date		Submission Date
Assessment type	Duration/Length of Assessment Type	Weighting of Assessment
Reports 3000 Words	End of the Module	100%

Learner declaration
I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Marks Awarded			
First assessor			
IV marks			
Agreed grade			
Signature of the assessor		Date	

FEEDBACK FORM
INTERNATIONAL COLLEGE OF BUSINESS & TECHNOLOGY

Module/Title: Database Design and Development /Design a Database Management System for
“TRAVEL CLASSICS” travels

Student:

Assessor:

Assignment:

Feedback:

Marks Awarded:

Contents

Task 1	5
Task 2	13
Task 3	17
Task 4	18
Task 5	23
Task 6	26
Task 7	41
Task 8	42
References	52

Task 1

1.1 Data models

The organization of data pieces within an information system is determined by a data model. A data model describes how data is accessed from storage and how it is related to other data items. Data models frequently use graphs or data model diagrams to show how data flow. Since business teams can identify the data and data formats required for business functions, and software teams can create the responses required for those requests, this visual depiction aids in communication between software and business teams.

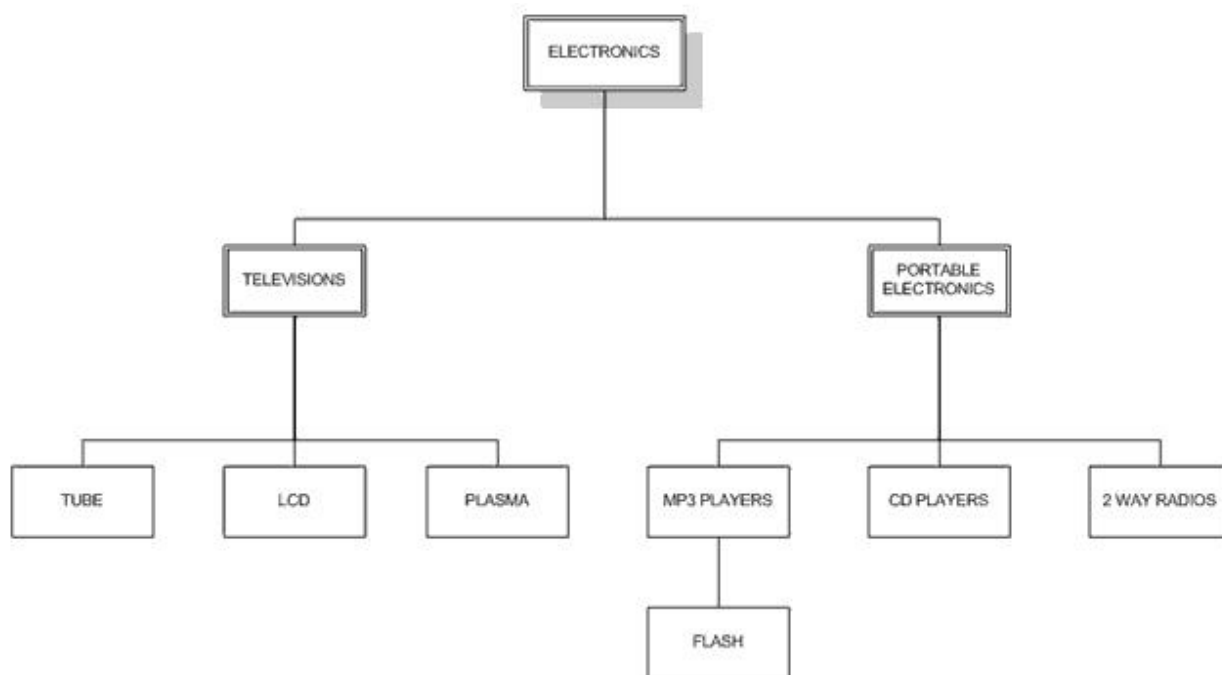
Type of data models.

- Hierarchical Data Model
- Network Model
- Entity-Relationship Model
- Relational Model
- Object-Oriented Data Model
- Object-Relational Data Model
- Dimensional Data Model
- Float Model
- Context Model
- Semi-structured data model

Hierarchical Data Model

The first database management system concept was the hierarchical model. The data in this idea is collected using a hierarchical tree structure. The root of the hierarchy, which includes the root data, is where it all starts. As child nodes are added to the parent node, the hierarchy develops into a tree. Food recipes, website sitemaps, and other real-world relationships are all depicted in this model with accuracy.

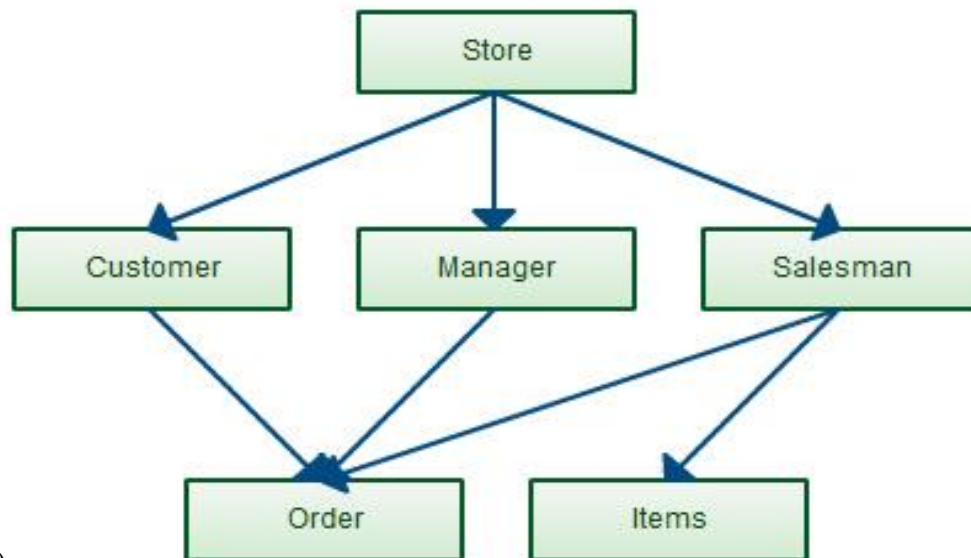
This model is an accurate representation of various relationships seen in the actual world, including those found in sitemaps, recipes, and other data. (Raveendran, 2023)



(Dadhich, n.d.)

Network Model

- In order to increase database performance and standards, the network model was developed to more efficiently describe complex data interactions as compared to hierarchical models.
- A graphical representation of its entities is used to organize them, and some of the entities have many paths for access. A user sees the network model as a grouping of records with 1:M relationships.
- Ability to Merge Relationships: -The data in this model is more closely related because there are more relationships. It has the capacity to control both one-to-one and many-to-many relationships.
- Many paths – Due to numerous relationships, there may be more than one way to reach the same record. It facilitates quick and easy data access.
- Circular Linked List – This model uses a circular linked list to carry out its activities. A software is used to retain the current position, which uses relationships to search through the entries. (Priya, 2021)

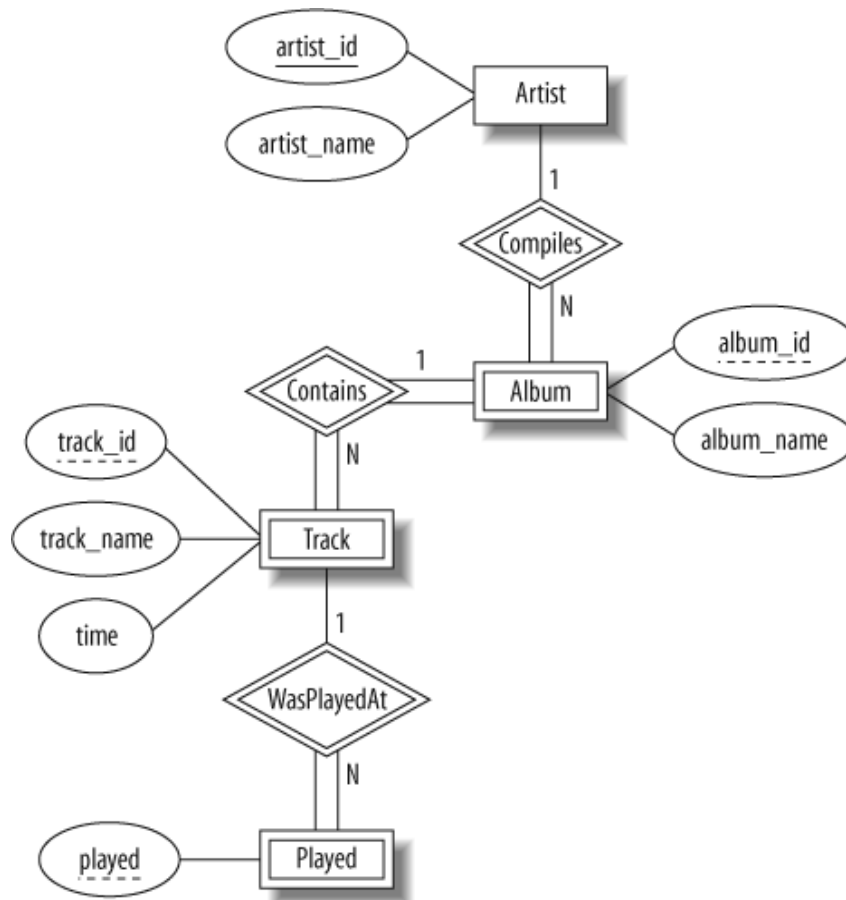


(Anon., 2022)

Entity-Relationship Model (ER Model)

Entity Relationship Model, or ER Model, is a high-level conceptual data model diagram. To create a well-designed database, rigorous analysis of the data requirements is assisted by the ER model. The ER Model depicts actual people, things, and the connections between them. Before deploying your database, it is recommended that you create an ER Model in a DBMS.

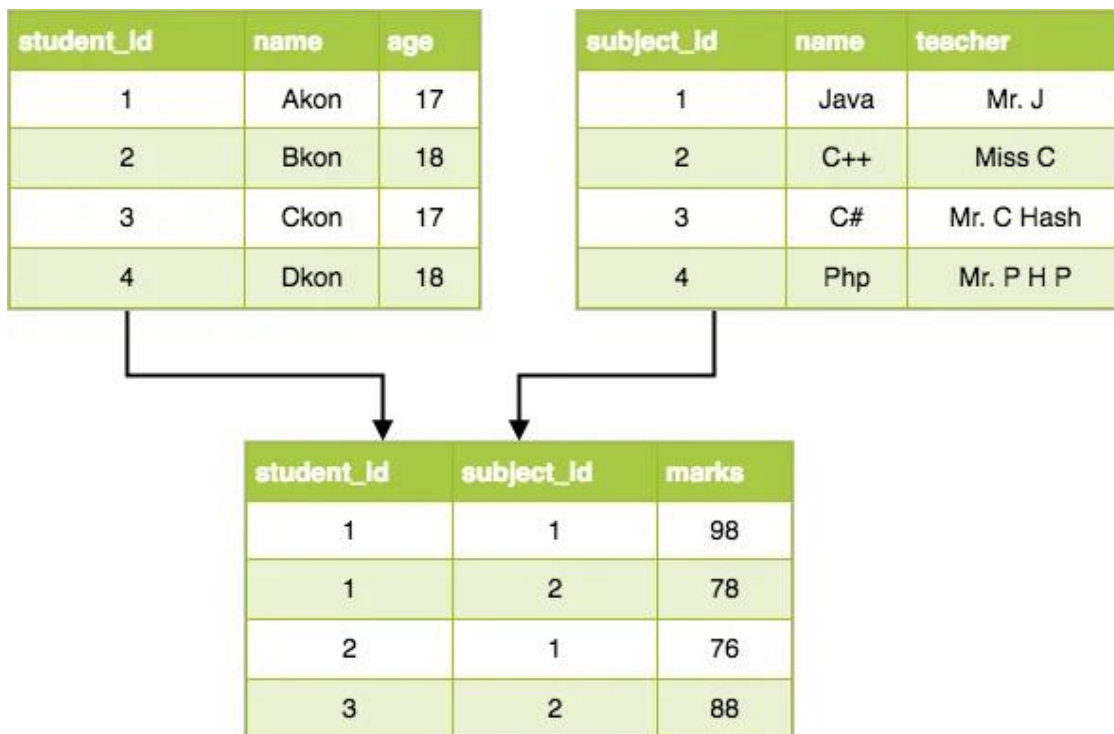
You can create a well-designed database by using ER Modeling to methodically assess your data requirements. Therefore, finishing ER modelling before installing your database is regarded as best practise. (Peterson, 2022)



Relational Model

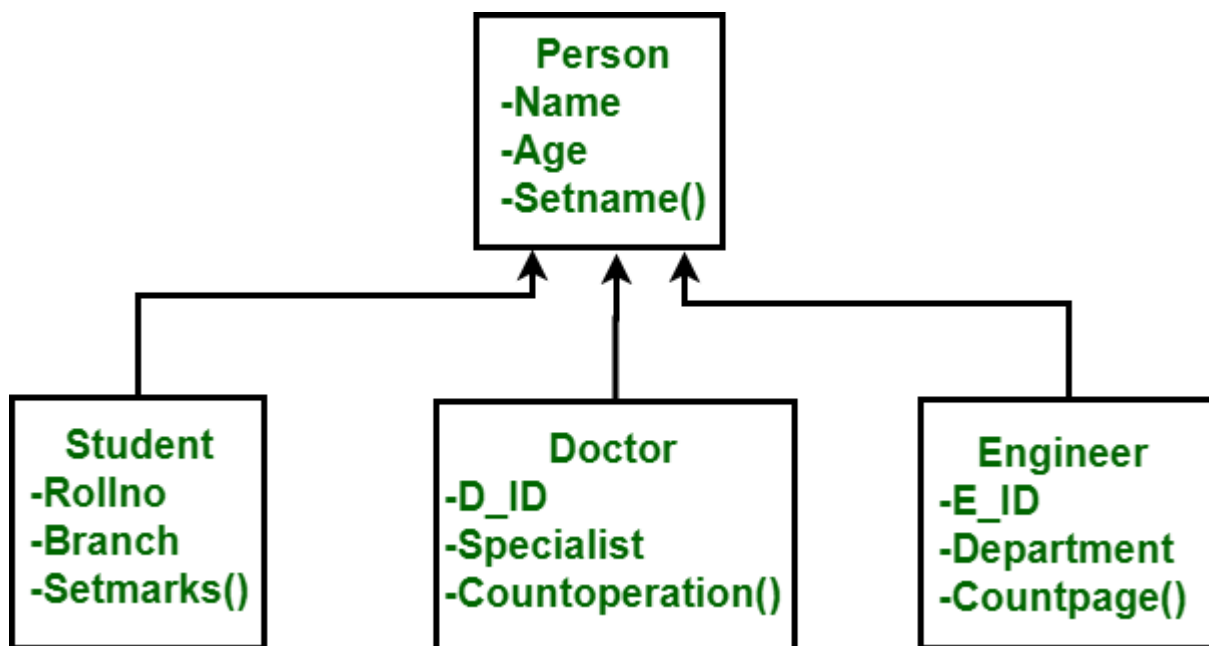
An technique to conceptually representing and managing the data kept in a database is the relational model for database management. The data in this model is arranged into a set of two-dimensional, mutually exclusive tables, or relations. Each relation is made up of a set of columns and rows, where the columns correspond to an entity's properties and the rows, or tuples, to its records.

A simple, effective, and adaptable method of storing and accessing structured data was made possible by the use of tables to store the data. This data model's simplicity makes it simple to access and sort the data. As a result, it is widely utilised for data processing and storage throughout the world. (Gulati, 2022)



Object -Oriented Model (OOM)

A collection of objects that include the stored values of the instance variables present within an object is used in an object-oriented model (OOM) to generate new objects. Models that are record-oriented contrast with object-oriented values, which only contain objects. The object-oriented modeling methodology unites the creation of applications and databases and transforms them into a single data model and linguistic environment. Data abstraction, inheritance, and encapsulation are all supported by object-oriented model, which enables object communication and identification. (Stoltzfus, 2023)



1.2 Critically compare the Difference between Hierarchical, Network, and Relational Data Models:

No.	Hierarchical Data Model	Network Data Model	Relational Data Model
1.	Hierarchical data storage is used in this architecture. Although it is the oldest, it is no longer in use.	Using links or pointers, it arranges records in relation to one another.	It arranges data in a table-like format, and relationships between tables are established via shared fields.
2.	It employs a tree structure to arrange the records.	Records are arranged using directed graphs.	It uses tables to organize records.
3.	The 1:1 and 1:n relations are implemented.	It also uses many-to-many relationships in addition to 1:1 and 1: n interactions.	It implements many to many relationships in addition to 1:1 and 1: n.
4.	To build physical linkages between records, pointers are utilized.	To establish a physical relationship between records, a linked list is employed.	Rows and columns are employed in the logical form to show the relationships between records.
5.	Update leads to inconsistency problems because of the existence of multiple instances of a record	There is no issue since there is just one instance of the records.	With the removal of redundant data during the normalization process, updating a record is quick and easy.
6.	Because of their complexity, databases are challenging to design.	Due to their complexity, databases are challenging to build and manage. It thereby places a load on the coder.	Due to the end user's ability to conceal physical level information, it is simple to understand.

(ankita_saini, 2022)

1.3 why new data models are replacing older data models

It's simply the progression of computing, as you can observe with, for instance, the development of the internet. When data volumes increase, outdated structures begin to lose efficiency, and businesses must learn how to enhance back-end services to meet customer demand. That's only one of the numerous aspects of data structures that might age or become redundant over time. (Sirichand, 2021)

Advantage of new data Models.

- Faster performance than older models
- Better and more clear documentation
- Fewer data, application errors.
- More secure.

Task 2

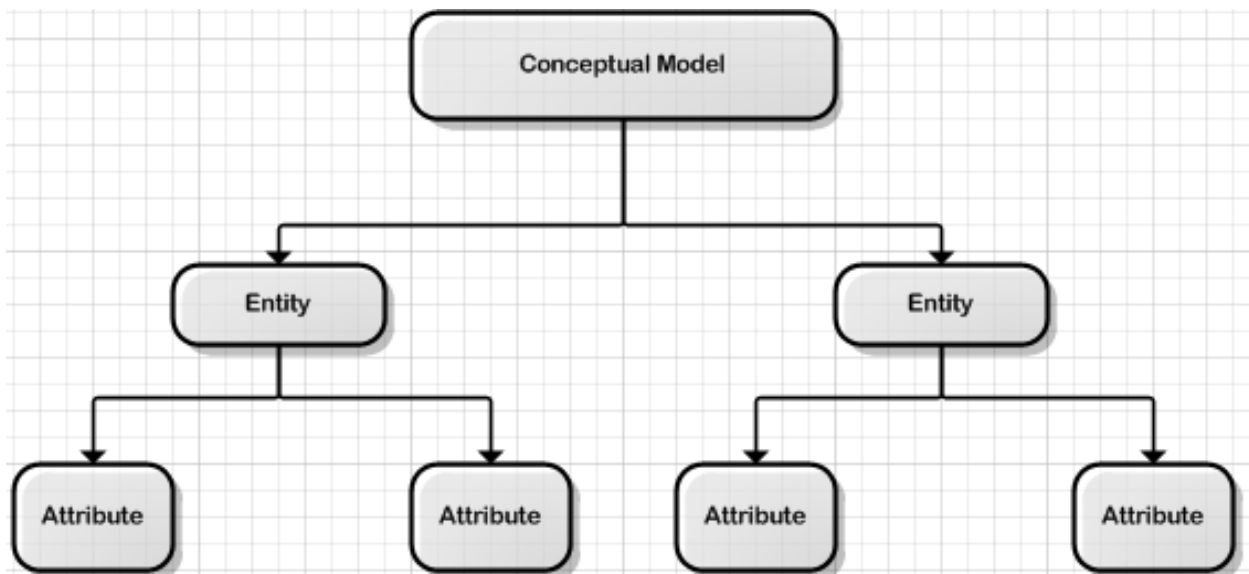
2 Differ Type of database design

2.1 Top-down design method

A top-down design is the breakdown of a system into smaller pieces in order to understand its constituent sub-systems.

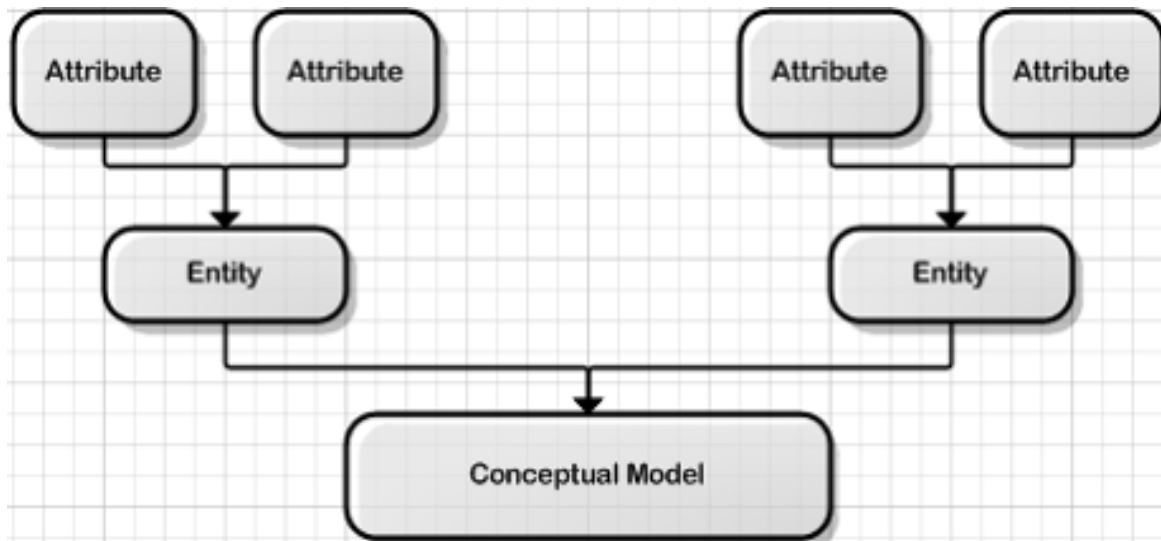
In top-down design, a system's overview is developed, specifying but without detailing any first-level subsystems. The specification is then broken down into its constituent parts by further refining each subsystem, for instance, by occasionally creating many levels of subsystems.

It is simpler to construct these underlying components as computer modules after they have been identified. Once the modules are constructed, assembling them to create the entire system is simple and quick. A top-down design is also referred to as a stepwise design. (SYNONYMS, 2014)



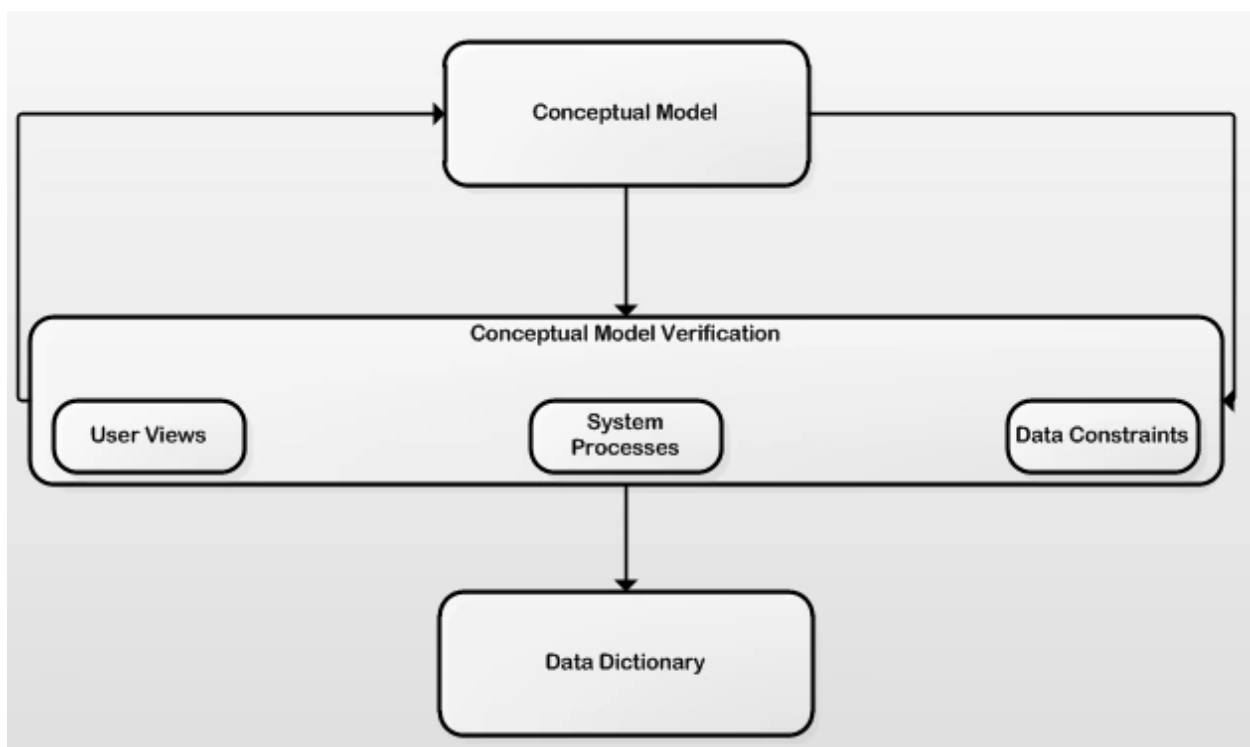
2.2 Bottom- up design method

Each component of the system is specifically described in this design. The smaller components are connected to create larger components, which are then connected to create a full system. A bottom-up method is used by object-oriented languages like C++ or java, where each object is recognized first. Make choices regarding reusable low-level utilities, and then choose how to combine them to create high-level constructs. (shivani7081, 2022)



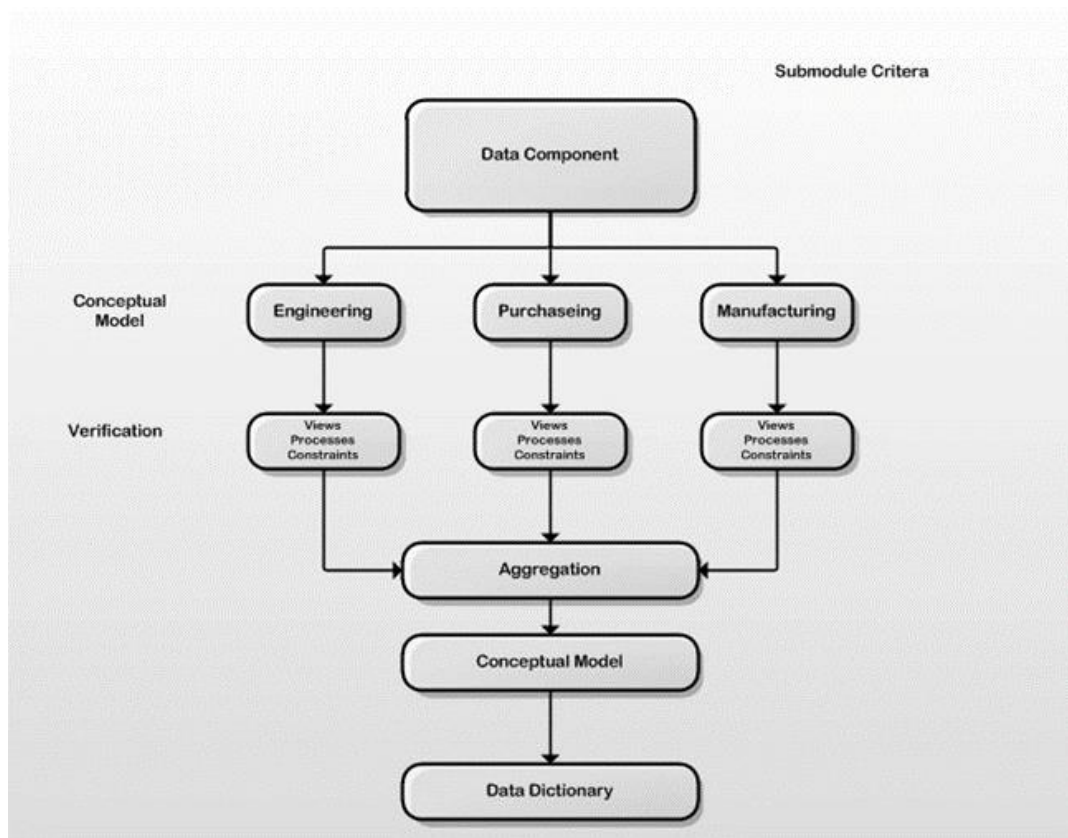
2.3 Centralized design

When the data component is made up of a relatively small number of objects and methods, a centralized design is most effective. A reasonably simple database can implement and reflect the design. A single database administrator or a small design team can successfully implement the centralized design, which is characteristic of basic or small databases. To make sure the design meets with the objectives of the organization, this individual or team will describe the problems, develop the conceptual design, validate the design using user perspectives, and specify system procedures and data restrictions. The centralized design, however, is not just used by small businesses. Even massive businesses can function in a straightforward database environment. (fandom, 2010)

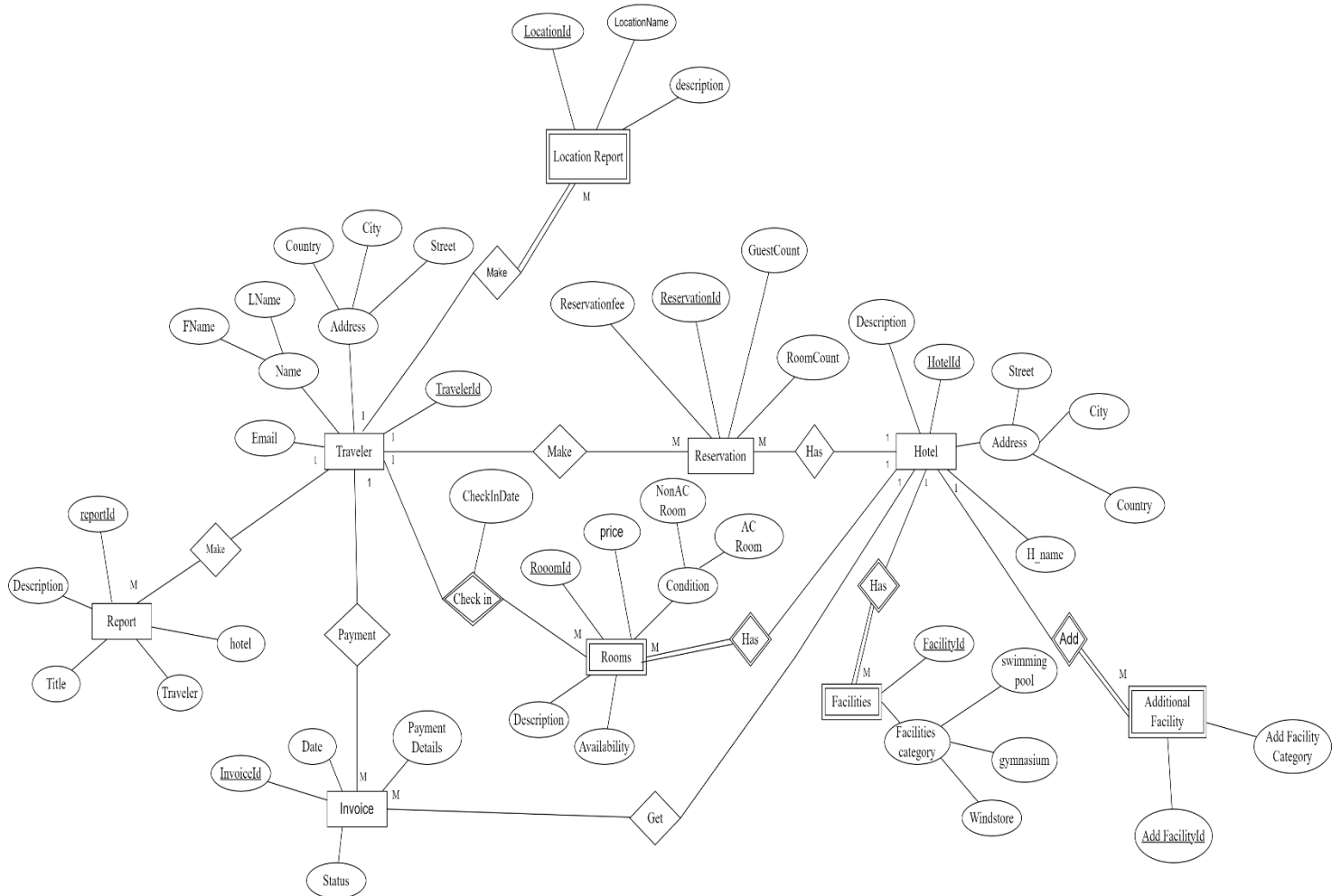


2.4 Decentralized design

- A distributed database is made up of two or more database files that are dispersed across numerous network nodes.
- The database is divided into multiple files, to put it simply.
- Users can access the nearest database file. Data retrieval is sped up as a result.
- The data that is relevant to them is also accessible to and controlled by a large number of people. This prevents users from interfering with one another.
- Another advantage is having backup database files in case the primary one fails. Users can still access them as a result, and the system is still operational.



Task 3



Task 4

Step 1 (Mapping of Regular Entity)

Traveler

<u>Traveler id</u>	Email	F_Name	L_Name	City	Street
--------------------	-------	--------	--------	------	--------

Hotel

<u>Hotel id</u>	Description	H_name	City	Street	Country
-----------------	-------------	--------	------	--------	---------

Reservation

<u>Reservation_id</u>	Reservation fee	guest-Count	Room-Count
-----------------------	-----------------	-------------	------------

Invoice

<u>Invoice_id</u>	Payment Details	Status	Date
-------------------	--------------------	--------	------

Report

<u>Report Id</u>	Hotel	Traveler	Title	Description
------------------	-------	----------	-------	-------------

Step 2 (Mapping of Weak Entity)

Traveler

<u>Traveler_ID</u>	Email	F_Name	L_Name	City	Street	Country
--------------------	-------	--------	--------	------	--------	---------

Hotel

<u>Hotel_Id</u>	Description	H_name	City	Street	Country
-----------------	-------------	--------	------	--------	---------

Room

<u>Room_id</u>	Price	Non Ac Room	AC Room	Description	Availability	Traveler_id -----	Hotel_id -----
----------------	-------	-------------	---------	-------------	--------------	----------------------	-------------------

Facilities

<u>Facility_id</u>	swimming pool	gymnasium	Windstore	Hotel_id -----
--------------------	---------------	-----------	-----------	-------------------

Location Report

<u>L_report ID</u>	Location	description	Traveler_id -----
--------------------	----------	-------------	----------------------

Additional Facility

Add Facility_Id	Add Facility Category	Hotel_id -----
-----------------	--------------------------	-------------------

Step 3 (Mapping of 1:1 Relationship)

There no 1:1 Relationship

Step 4 (Mapping of 1:M Relationship)

Traveler

<u>Traveler_ID</u>	Email	F_Name	L_Name	City	Street	Country
--------------------	-------	--------	--------	------	--------	---------

Hotel

<u>Hotel_Id</u>	Description	H_name	City	Street	Country
-----------------	-------------	--------	------	--------	---------

Reservation

<u>Reservation_id</u>	Reservation fee	guest-Count	Room-Count	Traveler_id -----	Hotel_id -----
-----------------------	-----------------	-------------	------------	----------------------	-------------------

Rooms

<u>Room_id</u>	Price	Non Ac Room	AC Room	Description	Availability	Hotel_id -----	Traveler_id -----
----------------	-------	-------------	---------	-------------	--------------	-------------------	----------------------

Facilities

<u>Facility_id</u>	swimming pool	gymnasium	Windstore	Hotel_id -----
--------------------	---------------	-----------	-----------	-------------------

Invoice

<u>Invoice_id</u>	Payment Details	Status	Date	Traveler_id -----	Hotel_id -----
-------------------	-----------------	--------	------	----------------------	-------------------

Report

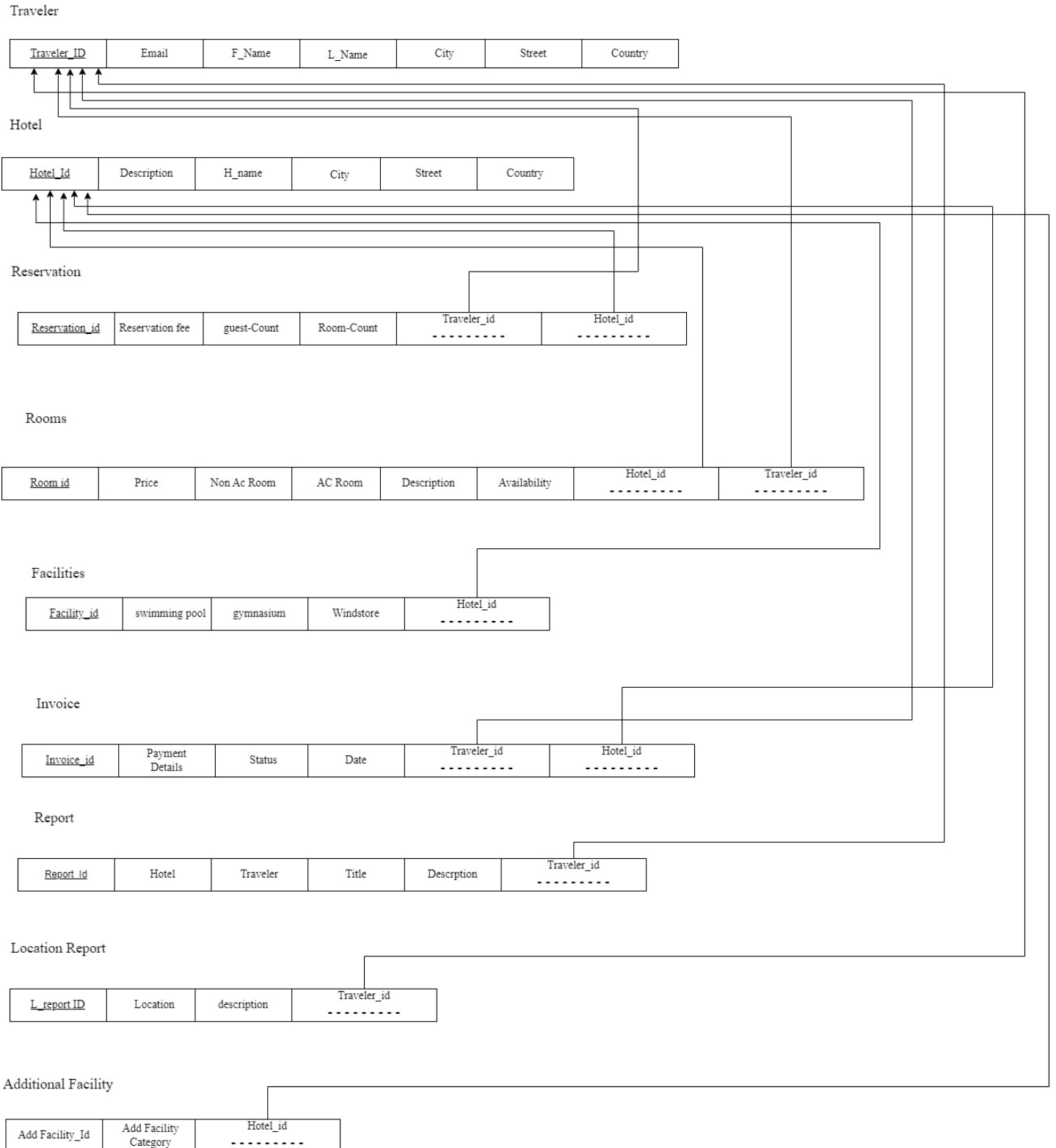
<u>Report_id</u>	Hotel	Traveler	Title	Description	Traveler_id -----
------------------	-------	----------	-------	-------------	----------------------

Location Report

<u>L_report_ID</u>	Location	description	Traveler_id -----
--------------------	----------	-------------	----------------------

Additional Facility

Add Facility_Id	Add Facility Category	Hotel_id -----
-----------------	-----------------------	-------------------



Step 5 (Mapping of M: N relationship)

There no 1:1 Relationship

Step 6 (Mapping of multivalued and composite attributes)

There no multivalued and composite attributes

Step 7 (Mapping of Isa relationship)

There no Isa relationship

Relational schema

Traveler

<u>Traveler_ID</u>	Email	F_Name	L_Name	City	Street	Country
--------------------	-------	--------	--------	------	--------	---------

Hotel

<u>Hotel_Id</u>	Description	H_name	City	Street	Country
-----------------	-------------	--------	------	--------	---------

Reservation

<u>Reservation_id</u>	Reservation fee	guest-Count	Room-Count	Traveler_id -----	Hotel_id -----
-----------------------	-----------------	-------------	------------	----------------------	-------------------

Rooms

<u>Room_id</u>	Price	Non Ac Room	AC Room	Description	Availability	Hotel_id -----	Traveler_id -----
----------------	-------	-------------	---------	-------------	--------------	-------------------	----------------------

Facilities

<u>Facility_id</u>	swimming pool	gymnasium	Windstore	Hotel_id -----
--------------------	---------------	-----------	-----------	-------------------

Invoice

<u>Invoice_id</u>	Payment Details	Status	Date	Traveler_id -----	Hotel_id -----
-------------------	--------------------	--------	------	----------------------	-------------------

Report

<u>Report_id</u>	Hotel	Traveler	Title	Description	Traveler_id -----
------------------	-------	----------	-------	-------------	----------------------

Location Report

<u>L_reportID</u>	Location	description	Traveler_id -----
-------------------	----------	-------------	----------------------

Additional Facility

Add Facility_Id	Add Facility Category	Hotel_id -----
-----------------	--------------------------	-------------------

Task 5

ER diagram Normalization

1NF (First Normal Form)

Rules

Each table cell should contain a single value.

Each record needs to be unique.

This Er diagram contains no have multi-valued attributes. There for already in 1NF.

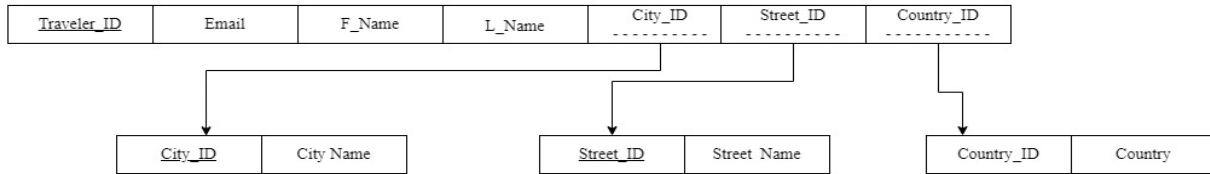
2 NF (Second Normal Form)

2NF (Second Normal Form) Rules

Rule 1- Be in 1NF

Rule 2- Single Column Primary Key that does not functionally dependent on any subset of candidate key relation.

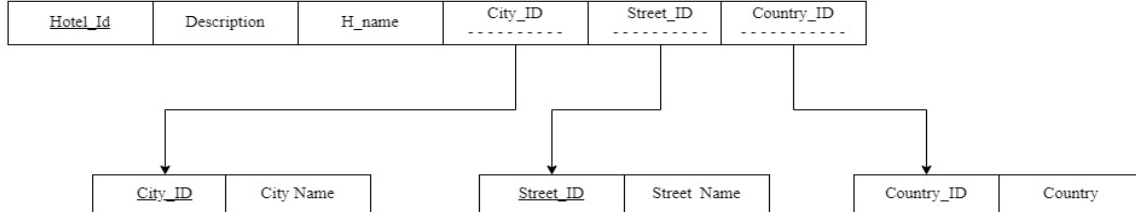
Traveler



City
Street
Country

} Not functionally dependant on
Traveler Primary Key

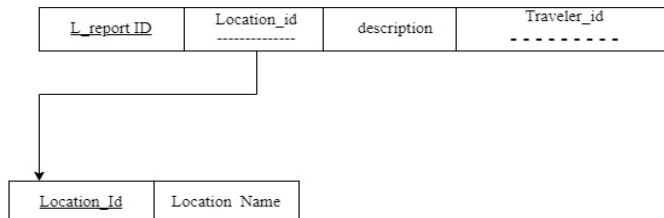
Hotel



City
Street
Country

} Not solely dependent on
Hotel Primary Key

Location Report



Location

} Not functionally dependent
on
Hotel Primary Key

3 NF (Third Normal Form)

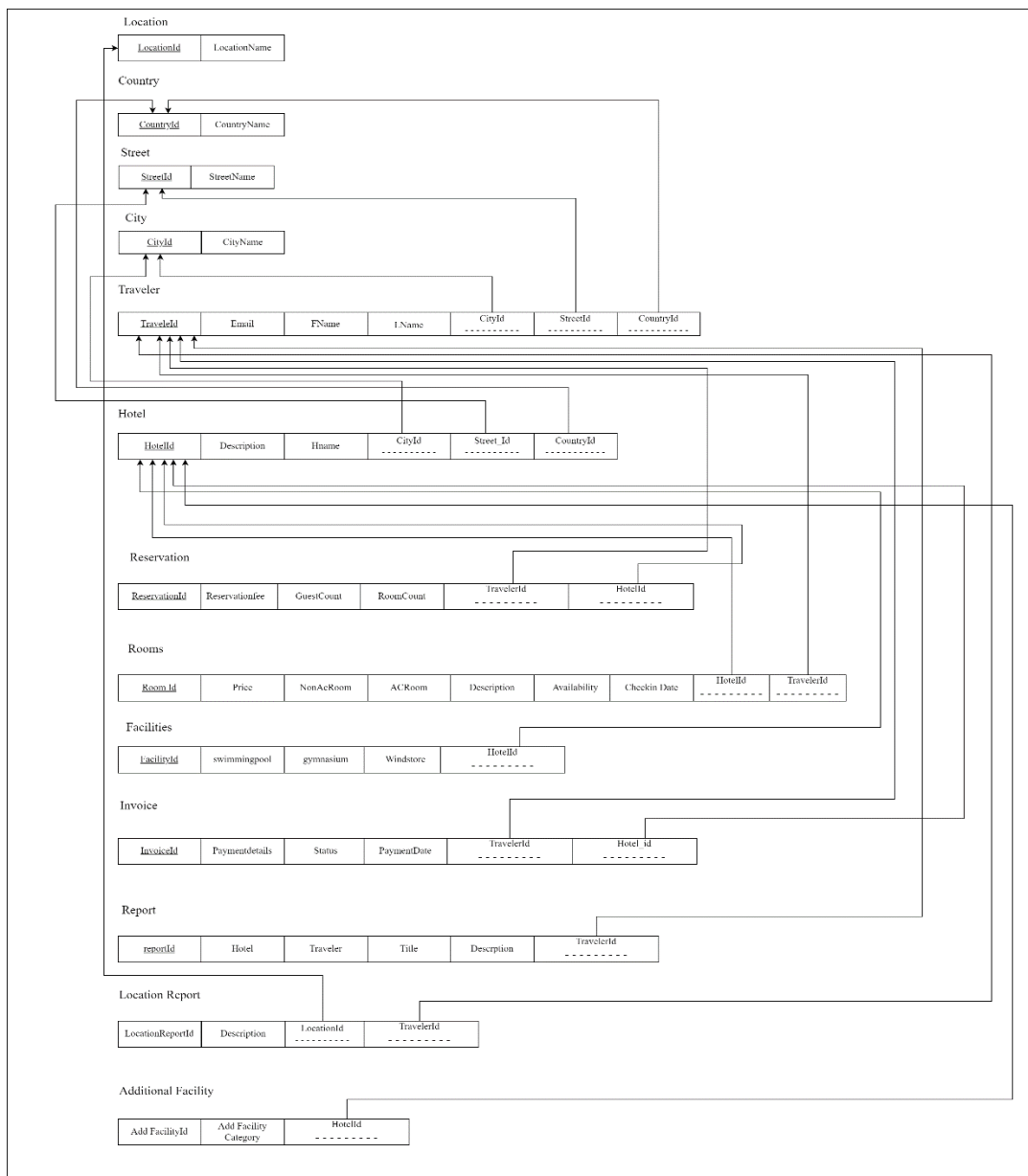
3NF (Third Normal Form) Rules

Rule 1- Be in 2NF

Rule 2- Has no transitive functional dependencies

This table is 3NF because there are no transitive functional relationships.

Final Relational schema



Task 6

Create database Travel_Classics

Use Travel_Classics

Create Table location(

 LocationId int not null,

 LocationName varchar(255),

 Primary Key (LocationId)

);

insert into location values(1,'Colombo')

insert into location values(2,'Sigiriya')

insert into location values(3,'Kandy')

insert into location values(4,'Galle')

insert into location values(5,'Nuwara Eliya')

insert into location values(6,'New york')

insert into location values(7,'Tokyo')

insert into location values (8,'London')

insert into location values(9,'Rio')

insert into location values(10,'Moscow')

insert into location values(11,'Doha')

insert into location values(12,'Queenstown')

insert into location values(13,'kerala')

insert into location values(14,'Miami')

insert into location values(15,'Abu Dhabi')

Select*from location

Create Table country (

CountryId int not null,

CountryName varchar (255),

Primary Key (CountryId)

);

insert into country values (1,'Sri Lanka')

insert into country values (2,'USA')

insert into country values (3,'Japan')

insert into country values (4,'Australia')

insert into country values (5,'England')

insert into country values (6,'Russia')

insert into country values (7,'Brazil')

insert into country values (8,'India')

insert into country values (9,'philipian')

insert into country values (10,'New Zealand')

insert into country values (11,'Thailand')

insert into country values (12,'China')

insert into country values (13,'Dubai')

insert into country values (14,'Canada')

insert into country values (15,'UAE')

select * from country

Create Table street(

```
StreetId int not null,  
StreetName varchar (255),  
Primary key(StreetId)  
  
);  
  
insert into street values (1,'Armour Street')  
insert into street values (2,'Fifth Avenue')  
insert into street values (3,'Takeshita Street')  
insert into street values (4,'Victoria Street')  
insert into street values (5,'Oxford Street')  
insert into street values (6,'Sovetskaya')  
insert into street values (7,'Avenida Rio Branco')  
insert into street values (8,'Broadway')  
insert into street values (9,'Ochard Road')  
insert into street values (10,'Saint Anthonys street')  
insert into street values (11,'Wolfendhal Street')  
insert into street values (12,'younge Street')  
insert into street values (13,'Robson Street')  
insert into street values (14,'king street')  
insert into street values (15,'john Street');
```

```
select * from street
```

```
Create Table city (
```

```
CityId int not null,
```

CityName varchar (255),

Primary key(CityId)

);

insert into city values (1,'Sri Jayewardanapura kotte')

insert into city values (2,'Washington')

insert into city values (3,'Osaka')

insert into city values (4,'Sydney')

insert into city values (5,'Liverpool')

insert into city values (6,'St.Petersburg')

insert into city values (7,'Sao Paulo')

insert into city values (8,'Shanghia')

insert into city values (9,'Berlin')

insert into city values (10,'San Francisco')

insert into city values (11,'Barcelona')

insert into city values (12,'Rome')

insert into city values (13,'Delhi')

insert into city values (14,'Bejjine')

insert into city values (15,'Torronto')

select * from city

Create Table traveler (

TravelerId int not null,

Email varchar (255),

FName varchar (255) not null,

```

LName varchar (255) not null,
CityId int not null,
StreetId int not null,
CountryId int not null,
Primary key(TravelerId),
Foreign key (CityId) References city (CityId),
Foreign key (StreetId) References street (StreetId),
Foreign key (CountryId) References country (CountryId)
);

insert into traveler values (1,'Abc@gmail.com','Abraham ','Lincoln',2,2,2)
insert into traveler values (2,'Bcs@gmail.com','Pasan','Chamathka',4,4,4)
insert into traveler values (3,'Rmb@gmail.com','sahan','Lahiru',6,7,8)
insert into traveler values (4,'Ght@gmail.com','Kasun','Kalhara',13,9,8)
insert into traveler values (5,'Rxy@gmail.com','Rashmika','Mandanna',10,5,2)
insert into traveler values (6,'Mcu@gmail.com','Minoli','Perera',12,10,7)
insert into traveler values (7,'pjr@gmail.com','pasindu','Melan',6,9,3)
insert into traveler values (8,'Rox@gmail.com','Yasiru','Nirmala',1,13,1)
insert into traveler values (9,'Rtx@gmail.com','Joel','Martine',5,8,2)
insert into traveler values (10,'Bbc@gmail.com','Abdul','Wahhab',14,1,7)
insert into traveler values (11,'Justin@gmail.com','Justin','Bieber',9,4,15)
insert into traveler values (12,'David@gmail.com','David','Guetta',9,12,6)
insert into traveler values (13,'Pcb@gmail.com','Bebe','Rexha',4,14,8)
insert into traveler values (14,'John@gmail.com','Johnny','depp',4,10,9)
insert into traveler values (15,'Mastros@gmail.com','George ','Mastros',12,10,7)

select*from traveler

```

Create Table hotel (

HotelId int not null,
HName varchar(255)not null,
Description varchar (255),
CityId int not null,
StreetId int not null,
CountryId int not null,
Primary key (HotelId),
Foreign key (CityId) References city (CityId),
Foreign key (StreetId) References street (StreetId),
Foreign key (CountryId) References country (CountryId)

);

insert into hotel values (1,'Shangri-La','Sample Description',1,2,6)
insert into hotel values (2,'Galadari Hotel','Sample Description',4,7,9)
insert into hotel values (3,'Mount Lavinia Hotel','Sample Description',15,11,10)
insert into hotel values (4,'Mandarina Hotel','Sample Description',1,5,12)
insert into hotel values (5,'The kingsbury Hotel','Sample Description',6,12,14)
insert into hotel values (6,'Cinnamon Grand Hotel','Sample Description',4,14,15)
insert into hotel values (7,'Amaya Lake Hotel','Sample Description',2,5,8)
insert into hotel values (8,'Marino Beach Hotel','Sample Description',7,11,13)
insert into hotel values (9,'Cinnamon Red Hotel','Sample Description',8,9,10)
insert into hotel values (10,'Avain Bentota Resort Hotel','Sample Description',4,2,3)
insert into hotel values (11,'The Plaza hotel','Sample Description',7,4,5)
insert into hotel values (12,'Taj hotel','Sample Description',9,15,6)
insert into hotel values (13,'Atlantis The plam Hotel ','Sample Description',1,7,12)
insert into hotel values (14,'Ritz-Carlton Hotel','Sample Description',9,5,2)

insert into hotel values (15,'Beverly Hills hotel','Sample Description',1,13,10)

select*from hotel

Create Table reservation(

ReservationId int not null,

Reservationfee float not null,

GuestCount int not null,

RoomCount int not null,

Reservationdate Date not null,

TravelerId int not null,

HotelId int not null,

Primary key(ReservationId),

Foreign key (TravelerId) References traveler (TravelerId),

Foreign key (HotelId) References hotel(HotelId)

);

insert into reservation values(1,2500.00,10,15,'20230225',4,1)

insert into reservation values(2,4500.00,15,11,'20230228',2,4)

insert into reservation values(3,1250.00,20,15,'20230301',3,3)

insert into reservation values(4,3267.00,22,20,'20230402',5,1)

insert into reservation values(5,2440.00,10,4,'20230425',1,5)

insert into reservation values(6,2340.00,7,3,'20230227',2,6)


```
insert into reservation values(7,2540.00,8,4,'20230115',3,7)
insert into reservation values(8,1240.00,6,3,'20230328',2,8)
insert into reservation values(9,1440.00,5,2,'20230224',10,9)
insert into reservation values(10,1345.00,6,4,'20230226',9,5)
insert into reservation values(11,2000.00,10,8,'20230226',7,5)
insert into reservation values(12,3500.00,15,10,'20230226',8,5)
insert into reservation values(13,2440.00,6,4,'20230226',3,5)
insert into reservation values(14,2440.00,6,4,'20230226',6,5)
insert into reservation values(15,2440.00,6,4,'20230226',2,5)
```

```
select*from reservation
```

Create Table rooms(

```
    RoomId int not null,
    Price float not null,
    NonAcRoom int not null,
    AcRoom int not null,
    Description varchar (255),
    Availablity int not null ,
    Checkin Date not null ,
    HotelId int not null,
    TravelerId int not null,
    Primary key (RoomId),
    Foreign key (HotelId) References hotel(HotelId),
    Foreign key (TravelerId) References traveler (TravelerId)
);
```

```

insert into rooms values(1,2500.00,1,0,'Sample Description',0,'20230227',4,3)
insert into rooms values(2,6720.00,0,1,'Sample Description',1,'20230527',5,2)
insert into rooms values(3,1250.00,1,0,'Sample Description',0,'20230312',2,3)
insert into rooms values(4,3500.00,0,1,'Sample Description',1,'20230226',1,3)
insert into rooms values(5,4750.00,1,0,'Sample Description',0,'20230327',2,4)
insert into rooms values(6,4325.00,0,1,'Sample Description',0,'20230325',6,7)
insert into rooms values(7,2500.00,1,0,'Sample Description',1,'20230317',7,14)
insert into rooms values(8,4350.00,0,1,'Sample Description',0,'20230220',8,4)
insert into rooms values(9,1200.00,1,0,'Sample Description',1,'20230321',6,10)
insert into rooms values(10,5320.00,0,1,'Sample Description',0,'20230214',12,9)
insert into rooms values(11,3450.00,1,0,'Sample Description',1,'20230217',7,14)
insert into rooms values(12,4500.00,0,1,'Sample Description',0,'20230218',8,13)
insert into rooms values(13,1234.00,1,0,'Sample Description',1,'20230214',5,11)
insert into rooms values(14,4320.00,0,1,'Sample Description',0,'20230216',12,14)
insert into rooms values(15,4440.00,1,0,'Sample Description',1,'20230222',7,8)

```

```
select*from rooms
```

Create Table facilites(

```

FacilityId int not null,
Swimmingpool int not null ,
Gymnasium int not null,
Windsotre int not null,
HotelId int not null,
Primary key (FacilityId),
Foreign key (HotelId) References hotel(HotelId)

```

);

insert into facilites values(1,0,1,1,3)

insert into facilites values(2,1,0,1,4)

insert into facilites values(3,0,1,1,5)

insert into facilites values(4,1,0,0,2)

insert into facilites values(5,0,1,0,4)

insert into facilites values(6,0,1,0,6)

insert into facilites values(7,0,1,0,7)

insert into facilites values(8,0,1,0,8)

insert into facilites values(9,0,1,0,9)

insert into facilites values(10,0,1,0,10)

insert into facilites values(11,0,1,0,12)

insert into facilites values(12,0,1,0,14)

insert into facilites values(13,0,1,0,8)

insert into facilites values(14,0,1,0,15)

insert into facilites values(15,0,1,0,111)

select*from facilites

Create Table invoice(

InvoiceId int not null,

Paymentdetails varchar (255),

Status varchar (255),

PaymentDate date not null,

TravelerId int not null,

HotelId int not null,

Primary key (InvoiceId),

Foreign key (TravelerId) References traveler (TravelerId),
Foreign key (HotelId) References hotel(HotelId)
);

insert into Invoice values (1,'Sample Paymentdetails','Incomplete','20230128',1,2)
insert into Invoice values (2,'Sample Paymentdetails','Advanced payed','20230215',3,4)
insert into Invoice values (3,'Sample Paymentdetails','Completed','20230301',2,5)
insert into Invoice values (4,'Sample Paymentdetails','Advanced payed','20230217',4,1)
insert into Invoice values (5,'Sample Paymentdetails','Completed','20230320',6,7)
insert into Invoice values (6,'Sample Paymentdetails','InCompleted','20230320',8,9)
insert into Invoice values (7,'Sample Paymentdetails','Advanced payed','20230320',10,3)
insert into Invoice values (8,'Sample Paymentdetails','InCompleted','20230320',14,7)
insert into Invoice values (9,'Sample Paymentdetails','Advanced payed','20230320',12,8)
insert into Invoice values (10,'Sample Paymentdetails','Completed','20230320',11,12)
insert into Invoice values (11,'Sample Paymentdetails','InCompleted','20230320',4,10)
insert into Invoice values (12,'Sample Paymentdetails','Advanced payed','20230320',6,12)
insert into Invoice values (13,'Sample Paymentdetails','InCompleted','20230320',7,15)
insert into Invoice values (14,'Sample Paymentdetails','Completed','20230320',5,3)
insert into Invoice values (15,'Sample Paymentdetails','Advanced
payed','20230320',13,14)

select*from Invoice

Create Table report(

ReportId int not null,
Hotel varchar (255),
Traveler varchar (255),
Title varchar (255),

```

Description varchar (255),
TravelerId int not null,
Primary key(ReportId),
Foreign key (TravelerId) References traveler (TravelerId)

);

insert into report values(1,'Shangri-La','Abraham','sample title','Sample Description ',1)
insert into report values(2,'Galadari Hotel','Pasan','sample title','Sample Description ',2)
insert into report values(3,'Mount Lavinia Hotel','sahan','sample title','Sample Description ',3)
insert into report values(4,'Mandarina Hotel','Kasun','sample title','Sample Description ',4)
insert into report values(5,'The kingsbury Hotel','Rashmika','sample title','Sample
Description ',5)
insert into report values(6,'Cinnamon Grand Hotel ','Minoli','sample title','Sample
Description ',6)
insert into report values(7,'Amaya Lake Hotel','pasindu ','sample title','Sample
Description ',7)
insert into report values(8,'Marino Beach Hotel',' Yasiru ','sample title','Sample
Description ',8)
insert into report values(9,'Cinnamon Red Hotel','Joel','sample title','Sample Description
',9)
insert into report values(10,'Avain Bentota Resort Hotel','Abdul','sample title','Sample
Description ',10)
insert into report values(11,'The Plaza hotel','Justin','sample title','Sample Description',11)
insert into report values(12,'Taj hotel ','David','sample title','Sample Description',12)
insert into report values(13,'Atlantis The plam Hotel','Bebe','sample title','Sample
Description',13)
insert into report values(14,'Ritz-Carlton Hotel ','Johnny','sample title','Sample
Description',14)
insert into report values(15,'Beverly Hills hotel',' George ','sample title','Sample
Description',15)

```

```
select*from report
```

```
Create Table locationreport(
```

```
    LocationreportId int not null,
```

```
    Description varchar (255),
```

```
    LocationId int not null,
```

```
    TravelerId int not null,
```

```
    Primary key(LocationreportId),
```

```
    Foreign key (LocationId) References location (LocationId),
```

```
    Foreign key (TravelerId) References traveler (TravelerId)
```

```
);
```

```
insert into locationreport values(1,'Sample Description',1,5)
```

```
insert into locationreport values(2,'Sample Description',2,4)
```

```
insert into locationreport values(3,'Sample Description',3,3)
```

```
insert into locationreport values(4,'Sample Description',4,2)
```

```
insert into locationreport values(5,'Sample Description',5,1)
```

```
insert into locationreport values(6,'Sample Description',5,4)
```

```
insert into locationreport values(7,'Sample Description',7,3)
```

```
insert into locationreport values(8,'Sample Description',8,6)
```

```
insert into locationreport values(9,'Sample Description',14,7)
```

```
insert into locationreport values(10,'Sample Description',15,9)
```

```
insert into locationreport values(11,'Sample Description',12,10)
```

```
insert into locationreport values(12,'Sample Description',11,14)
```

```
insert into locationreport values(13,'Sample Description',8,12)
```

insert into locationreport values(14,'Sample Description',4,6)

insert into locationreport values(15,'Sample Description',2,7)

select*from locationreport

Create Table additionalfacility(

AdditionalfacilityId int not null,

AddFacilitycategory varchar (255),

HotelId int not null,

Primary key (AdditionalfacilityId),

Foreign key (HotelId) References hotel(HotelId)

);

insert into additionalfacility values (1,'playground',2)

insert into additionalfacility values (2,'Badminton court',3)

insert into additionalfacility values (3,'tennis court',4)

insert into additionalfacility values (4,'Air rifle shoot',1)

insert into additionalfacility values (5,'volleyball court ',5)

insert into additionalfacility values (6,'Free wifi',8)

insert into additionalfacility values (7,'Spa',6)

insert into additionalfacility values (8,'SunSet boat trip',7)

insert into additionalfacility values (9,'outside catering service',10)

insert into additionalfacility values (10,'Car parking ',11)

insert into additionalfacility values (11,'dining rooms ',12)

insert into additionalfacility values (12,'Gift shop ',14)

insert into additionalfacility values (13,'Conference hall ',15)

```
insert into additionalfacility values (14,'Restaurant ',13)
```

```
insert into additionalfacility values (15,'Doctor on Call',10)
```

```
select*from additionalfacility
```


Task 7

- **SQL queries**

```
select t.TravelerId,t.Fname ,t.LName From traveler t,locationreport lr where lr.TravelerId=t.TravelerId  
And lr.LocationId = 1
```

```
select sum(Reservationfee) as tot from reservation where Reservationdate between '20230201' and  
'20230301' and HotelId = 5
```

```
select h.HName,r.RoomId,r.Availablity from Hotel h,rooms r where h.HotelId = r.HotelId
```

Task 8

Test Plan & Test Case

Test Plan

No.	Test case Name ID	Test case name	Scenario	Expected results
1	TC01	Test Column Name Problems	Checking the problems in Column Name	“Command (s) Complete Successfully”
2	TC02	test Column data types (Numeric vs. Int)	Checking the problem in Column data types	“Command (s) Complete Successfully”
3	TC03	test Table name problems	Checking the problems in Table name	“Command (s) Complete Successfully”
4	TC04	test Tables without a primary key	Checking whether There are tables without a primary key	“Command (s) Complete Successfully”
5	TC05	test Tables without any data	Checking whether There are tables without any data	“Command (s) Complete Successfully”

Test case - TC01

USE [Travel_Classics]

GO

ALTER PROCEDURE [SQLCop].[test Column Name Problems]

AS

BEGIN

SET NOCOUNT ON

DECLARE @Output VarChar(max)

SET @Output = "

SELECT @Output = @Output + TABLE_SCHEMA + '.' + TABLE_NAME + '.' +
COLUMN_NAME + Char(13) + Char(10)

FROM INFORMATION_SCHEMA.COLUMNS

WHERE COLUMN_NAME COLLATE SQL_LATIN1_GENERAL_CP1_CI_AI
LIKE '%[^a-z_\$]%'

And TABLE_SCHEMA <> 'tSQLt'

Order By TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME

If @Output > "

Begin

Set @Output = Char(13) + Char(10)

+ Char(13) + Char(10)

+ Char(13) + Char(10)

+ @Output

EXEC tSQLt.Fail @Output

End

END;

Test case - TC02

USE [Travel_Classics]

GO

ALTER PROCEDURE [SQLCop].[test Column data types (Numeric vs. Int)]

AS

BEGIN

SET NOCOUNT ON

Declare @Output VarChar(max)

Set @Output = "

Select @Output = @Output + ProblemItem + Char(13) + Char(10)

From (

SELECT TABLE_SCHEMA + '.' + TABLE_NAME + '.' +
COLUMN_NAME As ProblemItem

FROM INFORMATION_SCHEMA.COLUMNS C

WHERE C.DATA_TYPE IN ('numeric','decimal')

AND NUMERIC_SCALE = 0

AND NUMERIC_PRECISION <= 18

AND TABLE_SCHEMA <> 'tSQLt'

) As Problems

Order By ProblemItem

If @Output > "

Begin

Set @Output = Char(13) + Char(10)

+ 'For more information: '

+

'http://blogs.lessthandot.com/index.php/DataMgmt/DataDesign/sql-server-types-numeric-vs-int'

+ Char(13) + Char(10)

+ Char(13) + Char(10)

+ @Output

EXEC tSQLt.Fail @Output

End

END;

Test case - TC03

USE [Travel_Classics]

GO

ALTER PROCEDURE [SQLCop].[test Table name problems]

AS

BEGIN

SET NOCOUNT ON

DECLARE @Output VarChar(max)

DECLARE @AcceptableSymbols VARCHAR(100)

SET @AcceptableSymbols = '_\$'

SET @Output = "

SELECT @Output = @Output + TABLE_SCHEMA + '.' + TABLE_NAME + Char(13)
+ Char(10)

FROM INFORMATION_SCHEMA.TABLES

WHERE TABLE_NAME COLLATE SQL_LATIN1_GENERAL_CP1_CI_AI Like '%[^a-z'
+ @AcceptableSymbols + ']%'

AND TABLE_SCHEMA <> 'tSQLt'

ORDER BY TABLE_SCHEMA, TABLE_NAME

If @Output > "

Begin

Set @Output = Char(13) + Char(10)

+ Char(13) + Char(10)

+ Char(13) + Char(10)

+ @Output

EXEC tSQLt.Fail @Output

End

END;

Test case - TC04

```
USE [Travel_Classics]
```

```
GO
```

```
ALTER PROCEDURE [SQLCop].[test Tables without a primary key]
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON
```

```
    DECLARE @Output VarChar(max)
```

```
    SET @Output = "
```

```
        SELECT      @Output = @Output + su.name + '.' + AllTables.Name + Char(13) +  
Char(10)
```

```
        FROM (
```

```
            SELECT      Name, id, uid
```

```
            From  sysobjects
```

```
            WHERE      xtype = 'U'
```

```
        ) AS AllTables
```

```
        INNER JOIN sysusers su
```

```
            On AllTables.uid = su.uid
```

```
        LEFT JOIN (
```

```
            SELECT parent_obj
```



```

        From sysobjects
        WHERE xtype = 'PK'
        ) AS PrimaryKeys
        ON AllTables.id = PrimaryKeys.parent_obj
WHERE      PrimaryKeys.parent_obj Is Null
          AND su.name <> 'tSQLt'
ORDER BY su.name,AllTables.Name

If @Output > "
    Begin
        Set @Output = Char(13) + Char(10)

                                + Char(13) + Char(10)
                                + Char(13) + Char(10)
                                + @Output
        EXEC tSQLt.Fail @Output
    End
END;

```

Test case - TC05

```
USE [Travel_Classics]
```

```
GO
```

```
ALTER PROCEDURE [SQLCop].[test Tables without any data]
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON
```

```
    DECLARE @Output VarChar(max)
```

```
    SET @Output = "
```

```
    CREATE TABLE #EmptyTables(Table_Name VarChar(100))
```

```
    EXEC sp_MSforeachtable 'IF NOT EXISTS(SELECT 1 FROM ?) INSERT INTO  
#EmptyTables VALUES("?")'
```

```
    SELECT      @Output = @Output + Table_Name + Char(13) + Char(10)
```

```
    FROM #EmptyTables
```

```
    Where Left(Table_Name, 7) <> '[tSQLt]'
```

```
    ORDER BY Table_Name
```

```
    DROP TABLE #EmptyTables
```

```
    If @Output > "
```

Begin

Set @Output = Char(13) + Char(10)

+ Char(13) + Char(10)

+ Char(13) + Char(10)

+ @Output

EXEC tSQLt.Fail @Output

End

END;

References

BYJUS. 2023. Hierarchical Model in DBMS | GATE Notes. [ONLINE] Available at: <https://byjus.com/gate/hierarchical-model-in-dbms-notes/#:~:text=of%20Hierarchical%20Model-,What%20is%20the%20Hierarchical%20Model%20in%20DBMS%3F,added%20to%20the%20parent%20node..> [Accessed 24 February 2023].

Hierarchical Model - Mr. Ashish's Classroom. 2023. Hierarchical Model - Mr. Ashish's Classroom. [ONLINE] Available at: <https://sites.google.com/site/aishshub/database-management-systems/hierarchical-model>. [Accessed 24 February 2023].

Explain the Network Model in DBMS . 2023. Explain the Network Model in DBMS . [ONLINE] Available at: <https://www.tutorialspoint.com/explain-the-network-model-in-dbms>. [Accessed 24 February 2023].

Guru99. 2023. Entity Relationship (ER) Diagram Model with DBMS Example. [ONLINE] Available at: <https://www.guru99.com/er-diagram-tutorial-dbms.html>. [Accessed 24 February 2023].

36%	29%	16%	24%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	forum.red-gate.com Internet Source	6%
2	Submitted to University of Wales Institute, Cardiff Student Paper	4%
3	Submitted to University of Westminster Student Paper	3%
4	Submitted to University of Greenwich Student Paper	3%
5	www.simple-talk.com Internet Source	3%
6	Submitted to University of Wales, Bangor Student Paper	2%
7	www.coursehero.com Internet Source	2%
8	Submitted to University of Maryland, University College Student Paper	2%
9	Submitted to University of Bolton	