MSIS537-DATA MANAGEMENT 1

Final project report

Drinking Water Distribution Management

Swetha Adike

1. ABSTRACT

Water distribution system consists of an infrastructure that collect, treat, store and distribute water between sources and households. It is a combination of water sources, intake structures and service reservoirs through which water is distributed to consumers by a web of pipeline. To make it understandable, each component of water channels will be explained. The water distribution process consists of a huge database where there is a need to store data of each component on a daily basis or twice in a day, depending on the usage of water by customers. In this project, a logical database is created and engineered to relational data base and cardinalities are drawn between all the tables. Data is then inserted into these tables after importing the DDL file from Oracle SQL modeler to Oracle SQL Developer. Testing is also done using few SQL query commands to retrieve the data.

2. OBJECTIVE

The main objective of the project is to develop a relational database for drinking water distribution system which stores and keeps track of the information regarding the water sources, distribution structures, service reservoirs, household connections and pipelines in a state using Oracle SQL Modeler and Oracle SQL Developer. Aim of this project is to retrieve the information regarding the water sources in a habitation and also get information of network system, like from which structure water is distributed to a particular habitation or a household connection in a habitation. Also to draw information regarding the pipelines, in which a service reservoir is serving in a habitation

3. EXECUTIVE SUMMARY

Water distribution system carries water from a centralized water sources to a treatment plant. The portable water is then supplied to the consumers after treatment through network of pipes. The information regarding the water availability in each water distribution system component and supply to the households is maintained in digital format and physical format in each area, for day to day reference to release water to the consumers. This information in digital format is maintained either in SCADA system or in online or both.

Water distribution management, in this project, is a database management which refers to all water distribution system frameworks in a state, under taken by a government allied department, Mission Bhagiratha Department. The information is monitored and utilized by officers in the department at various levels to enhance the maintenance operations of all structures. The database consists of various water distribution system components, used in this project, in the order of the deliver water to a household are: Segments, Sources, Structures, and Service reservoirs.

Before going into the definitions of these components, there are few geographical divisions by which a State on a whole is divided. Also there are few terms which are to be explained to understand this project better, they are: District, Mandal, and Habitation. A State is divided into few **Districts** and these districts are subdivided into **Mandals**. Each Mandal is a group of **Habitations** put together. Each habitation consists of group of **households** which are customers, to which water is supplied by pipeline **network**. A household connection is a tap connection given to that household.

Segment: Depending upon the water availability, and irrespective of districts, a State is again divided into few segments. The division of segments is such that the geographical area fed by a particular river or lake comes under one segment. This helped the project in reducing the cost on the overall project.

Sources: Each river or lake watered to a segment is denoted as a source.

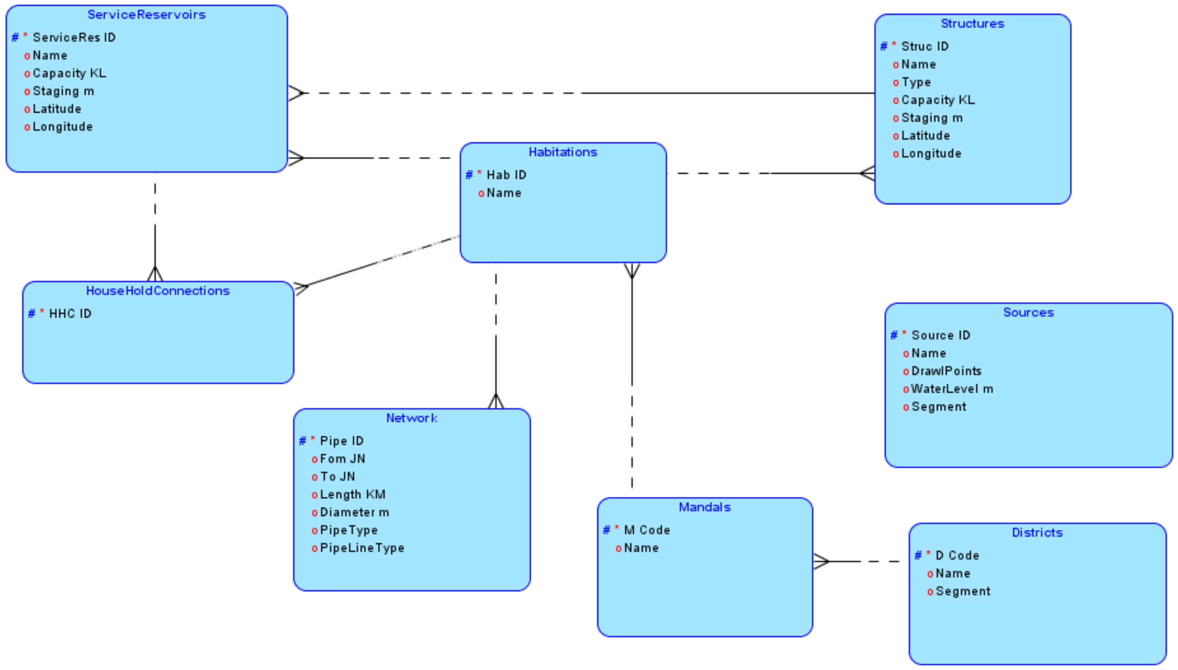
Structure: The water from source has to be stored temporarily in a structure for treating the water. The structure component has various types like intake wells, water treatment plants, break pressure tanks, and over head balancing reservoirs. Each of these is a temporary storage house for water to be treated or for treated water.

Service Reservoir: Water from these structures is stored in these service reservoirs or over head tanks, which is the water ready for supplying.

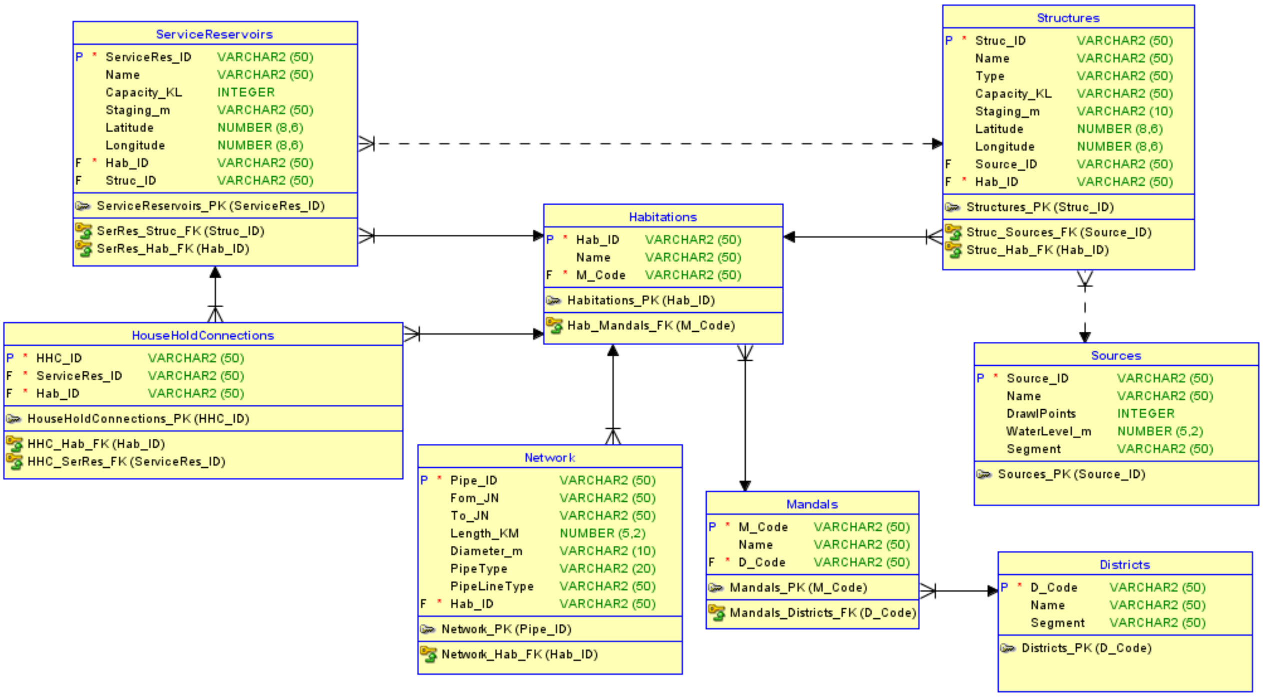
After relating all the above explained components in tables with respective fields, the cardinalities will explain the flow of the data in this database. The database also includes latitude and longitude to locate a structure component. By creating this database, quick information of structures and its allied service reservoir information can be drawn. Also, the relationship of a structure with a pipeline network can be extracted. All the possible complex queries were shown as an output to query in section 6.

4. ENTITY RELATIONSHIP (ER) DIAGRAM

The table entities are first created in logical data model in Oracle SQL data modeler as the figure below:



Then the above are engineered to relational data model in Oracle SQL data modeler as the figure below



5. TABLES

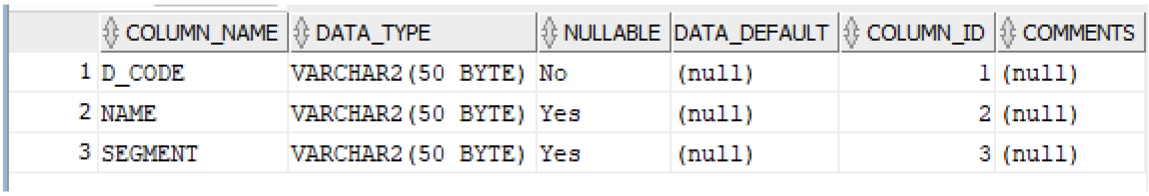
First, a table is created in the Oracle SQL data modeler and the fields are entered with their respective data types. After creating all tables, cardinal relationships are developed between the tables depending upon functional dependency. The details of the tables in above ER diagram and their cardinalities are given below.

a. District Table

i) Create Statement

```
CREATE TABLE districts (
    d_code      VARCHAR2(50) NOT NULL,
    name        VARCHAR2(50),
    segment     VARCHAR2(50)
);
```

ii) Table Result



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	D_CODE	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	NAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	SEGMENT	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)

```
ALTER TABLE districts ADD CONSTRAINT districts_pk PRIMARY KEY ( d_code );
```

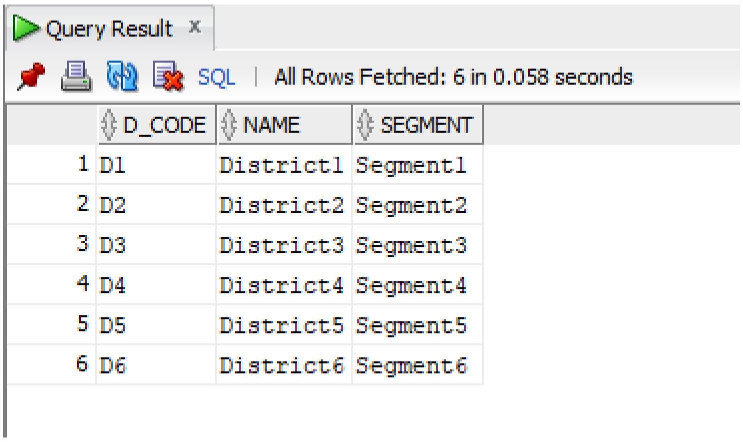
iii) Data Insertion

```
INSERT INTO "DISTRICTS" (D_CODE, NAME, SEGMENT) VALUES ('D1', 'District1', 'Segment1')
INSERT INTO "DISTRICTS" (D_CODE, NAME, SEGMENT) VALUES ('D2', 'District2', 'Segment2')
INSERT INTO "DISTRICTS" (D_CODE, NAME, SEGMENT) VALUES ('D3', 'District3', 'Segment3')
INSERT INTO "DISTRICTS" (D_CODE, NAME, SEGMENT) VALUES ('D4', 'District4', 'Segment4')
INSERT INTO "DISTRICTS" (D_CODE, NAME, SEGMENT) VALUES ('D5', 'District5', 'Segment5')
INSERT INTO "DISTRICTS" (D_CODE, NAME, SEGMENT) VALUES ('D6', 'District6', 'Segment6')
```

iv) Functional Dependency

D_Code →Name, Segment

v) Sample Data



SQL | All Rows Fetched: 6 in 0.058 seconds

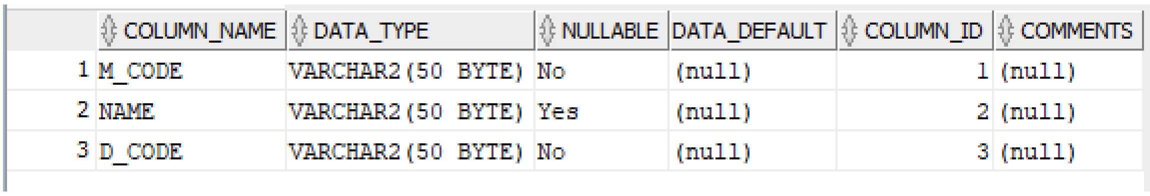
	D_CODE	NAME	SEGMENT
1	D1	District1	Segment1
2	D2	District2	Segment2
3	D3	District3	Segment3
4	D4	District4	Segment4
5	D5	District5	Segment5
6	D6	District6	Segment6

b. Mandal Table

i) Create Statement

```
CREATE TABLE mandals (
    m_code      VARCHAR2(50) NOT NULL,
    name        VARCHAR2(50),
    d_code      VARCHAR2(50) NOT NULL
);
```

ii) Table Result



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	M_CODE	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	NAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	D_CODE	VARCHAR2(50 BYTE)	No	(null)	3	(null)

```
ALTER TABLE mandals ADD CONSTRAINT mandals_pk PRIMARY KEY ( m_code );
ALTER TABLE mandals
    ADD CONSTRAINT mandals_districts_fk FOREIGN KEY ( d_code )
        REFERENCES districts ( d_code );
```

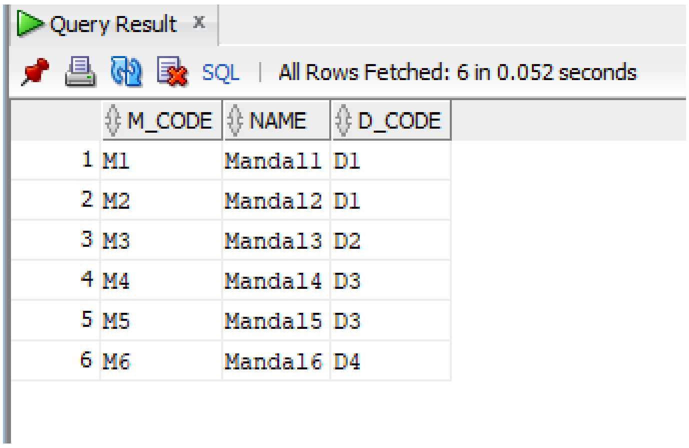
iii) Functional Dependency

M_Code →Name

iv) Data Insertion

```
INSERT INTO "MANDALS" (M_CODE, NAME, D_CODE) VALUES ('M1', 'Mandal1', 'D1')
INSERT INTO "MANDALS" (M_CODE, NAME, D_CODE) VALUES ('M2', 'Mandal2', 'D1')
INSERT INTO "MANDALS" (M_CODE, NAME, D_CODE) VALUES ('M3', 'Mandal3', 'D2')
INSERT INTO "MANDALS" (M_CODE, NAME, D_CODE) VALUES ('M4', 'Mandal4', 'D3')
INSERT INTO "MANDALS" (M_CODE, NAME, D_CODE) VALUES ('M5', 'Mandal5', 'D3')
INSERT INTO "MANDALS" (M_CODE, NAME, D_CODE) VALUES ('M6', 'Mandal6', 'D4')
```

v) Sample Data



All Rows Fetched: 6 in 0.052 seconds

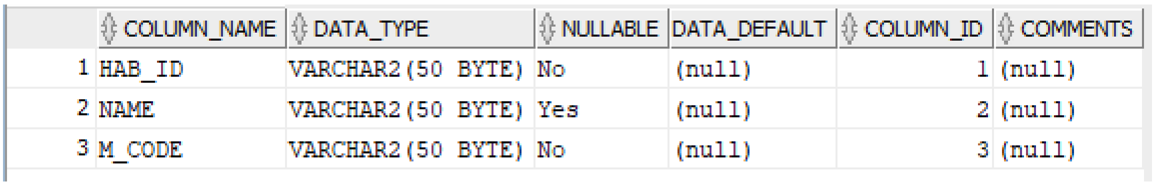
	M_CODE	NAME	D_CODE
1	M1	Mandal1	D1
2	M2	Mandal2	D1
3	M3	Mandal3	D2
4	M4	Mandal4	D3
5	M5	Mandal5	D3
6	M6	Mandal6	D4

c. Habitation Table

i) Create Statement

```
CREATE TABLE habitations (
    hab_id      VARCHAR2(50) NOT NULL,
    name        VARCHAR2(50),
    m_code      VARCHAR2(50) NOT NULL
);
```

ii) Table Result



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	HAB_ID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	NAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	M_CODE	VARCHAR2(50 BYTE)	No	(null)	3	(null)

```
ALTER TABLE habitations ADD CONSTRAINT habitations_pk PRIMARY KEY ( hab_id );
ALTER TABLE habitations
    ADD CONSTRAINT hab_mandals_fk FOREIGN KEY ( m_code )
        REFERENCES mandals ( m_code );
```

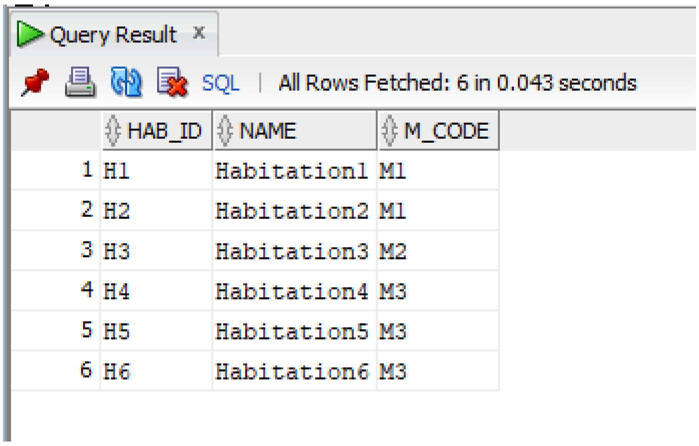
iii) Functional Dependency

Hab_ID→Name

iv) Data Insertion

```
INSERT INTO "HABITATIONS" (HAB_ID, NAME, M_CODE) VALUES ('H1', 'Habitation1', 'M1')
INSERT INTO "HABITATIONS" (HAB_ID, NAME, M_CODE) VALUES ('H2', 'Habitation2', 'M1')
INSERT INTO "HABITATIONS" (HAB_ID, NAME, M_CODE) VALUES ('H3', 'Habitation3', 'M2')
INSERT INTO "HABITATIONS" (HAB_ID, NAME, M_CODE) VALUES ('H4', 'Habitation4', 'M3')
INSERT INTO "HABITATIONS" (HAB_ID, NAME, M_CODE) VALUES ('H5', 'Habitation5', 'M3')
INSERT INTO "HABITATIONS" (HAB_ID, NAME, M_CODE) VALUES ('H6', 'Habitation6', 'M3')
```

v) Sample Data



All Rows Fetched: 6 in 0.043 seconds

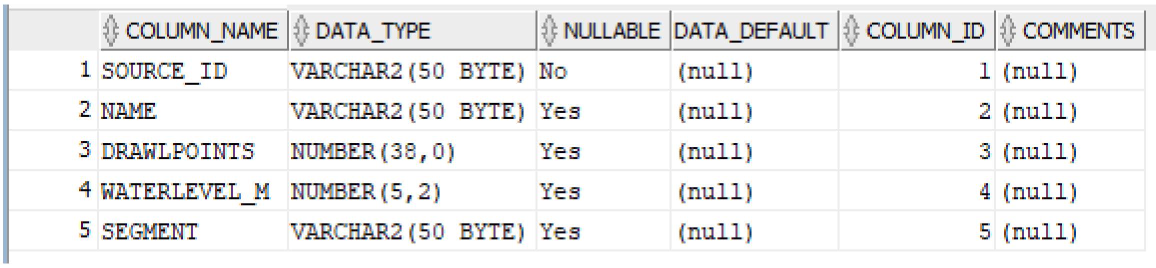
	HAB_ID	NAME	M_CODE
1	H1	Habitation1	M1
2	H2	Habitation2	M1
3	H3	Habitation3	M2
4	H4	Habitation4	M3
5	H5	Habitation5	M3
6	H6	Habitation6	M3

d. Source Table

i) Create Statement

```
CREATE TABLE sources (
    source_id           VARCHAR2(50) NOT NULL,
    name                VARCHAR2(50),
    drawlpoints         INTEGER,
    waterlevel_m        NUMBER(5, 2),
    segment             VARCHAR2(50)
);
```

ii) Table Result



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SOURCE_ID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	NAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	DRAWLPOINTS	NUMBER(38,0)	Yes	(null)	3	(null)
4	WATERLEVEL_M	NUMBER(5,2)	Yes	(null)	4	(null)
5	SEGMENT	VARCHAR2(50 BYTE)	Yes	(null)	5	(null)

ALTER TABLE sources ADD CONSTRAINT sources_pk PRIMARY KEY (source_id);

iii) Functional Dependency

Source_ID→Name, DrawlPoints, WaterLevel_m, Segment

iv) Sample Data Insertion

INSERT INTO "SOURCES" (SOURCE_ID, NAME, DRAWLPOINTS, WATERLEVEL_M, SEGMENT) VALUES ('S1', 'Source1', '2', '128.56', 'Segment1')

INSERT INTO "SOURCES" (SOURCE_ID, NAME, DRAWLPOINTS, WATERLEVEL_M, SEGMENT) VALUES ('S2', 'Source2', '1', '60.25', 'Segment2')

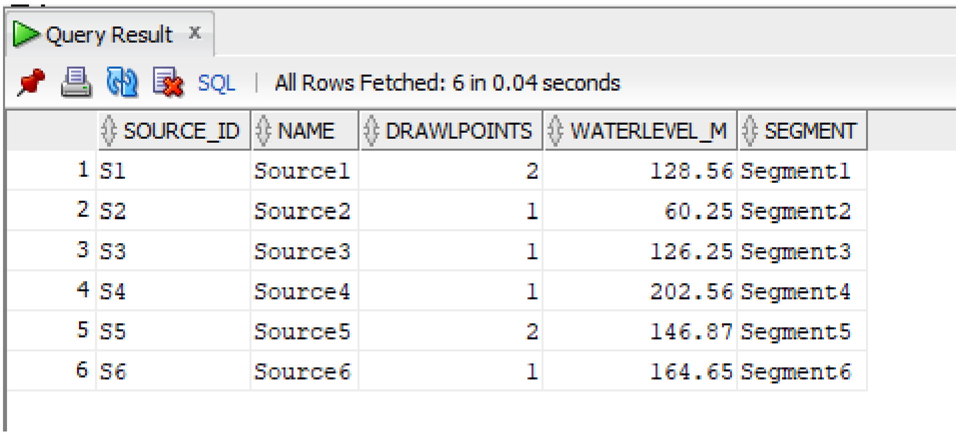
INSERT INTO "SOURCES" (SOURCE_ID, NAME, DRAWLPOINTS, WATERLEVEL_M, SEGMENT) VALUES ('S3', 'Source3', '1', '126.25', 'Segment3')

INSERT INTO "SOURCES" (SOURCE_ID, NAME, DRAWLPOINTS, WATERLEVEL_M, SEGMENT) VALUES ('S4', 'Source4', '1', '202.56', 'Segment4')

INSERT INTO "SOURCES" (SOURCE_ID, NAME, DRAWLPOINTS, WATERLEVEL_M, SEGMENT) VALUES ('S5', 'Source5', '2', '146.87', 'Segment5')

INSERT INTO "SOURCES" (SOURCE_ID, NAME, DRAWLPOINTS, WATERLEVEL_M, SEGMENT) VALUES ('S6', 'Source6', '1', '164.65', 'Segment6')

v) Sample Data Table



All Rows Fetched: 6 in 0.04 seconds

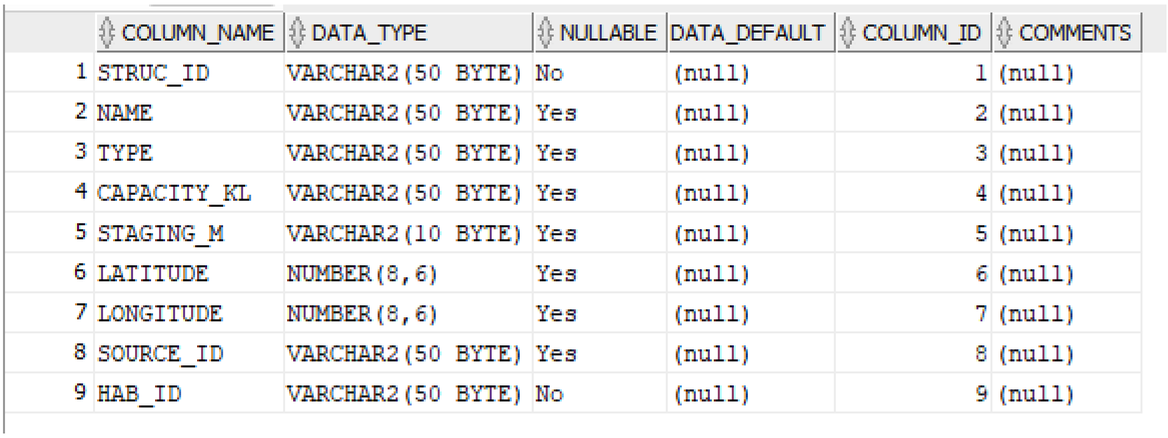
	SOURCE_ID	NAME	DRAWLPOINTS	WATERLEVEL_M	SEGMENT
1	S1	Source1	2	128.56	Segment1
2	S2	Source2	1	60.25	Segment2
3	S3	Source3	1	126.25	Segment3
4	S4	Source4	1	202.56	Segment4
5	S5	Source5	2	146.87	Segment5
6	S6	Source6	1	164.65	Segment6

e. Structure Table

i) Create Statement

```
CREATE TABLE structures (
    struc_id      VARCHAR2(50) NOT NULL,
    name          VARCHAR2(50),
    type          VARCHAR2(50),
    capacity_kl   VARCHAR2(50),
    staging_m     VARCHAR2(10),
    latitude      NUMBER(8, 6),
    longitude     NUMBER(8, 6),
    source_id     VARCHAR2(50),
    hab_id        VARCHAR2(50) NOT NULL
);
```

ii) Table Result



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	STRUC_ID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	NAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	TYPE	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	CAPACITY_KL	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	STAGING_M	VARCHAR2(10 BYTE)	Yes	(null)	5	(null)
6	LATITUDE	NUMBER(8,6)	Yes	(null)	6	(null)
7	LONGITUDE	NUMBER(8,6)	Yes	(null)	7	(null)
8	SOURCE_ID	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)
9	HAB_ID	VARCHAR2(50 BYTE)	No	(null)	9	(null)

```
ALTER TABLE structures ADD CONSTRAINT structures_pk PRIMARY KEY ( struc_id );
ALTER TABLE structures
    ADD CONSTRAINT struc_hab_fk FOREIGN KEY ( hab_id )
        REFERENCES habitations ( hab_id );

ALTER TABLE structures
    ADD CONSTRAINT struc_sources_fk FOREIGN KEY ( source_id )
        REFERENCES sources ( source_id );
```

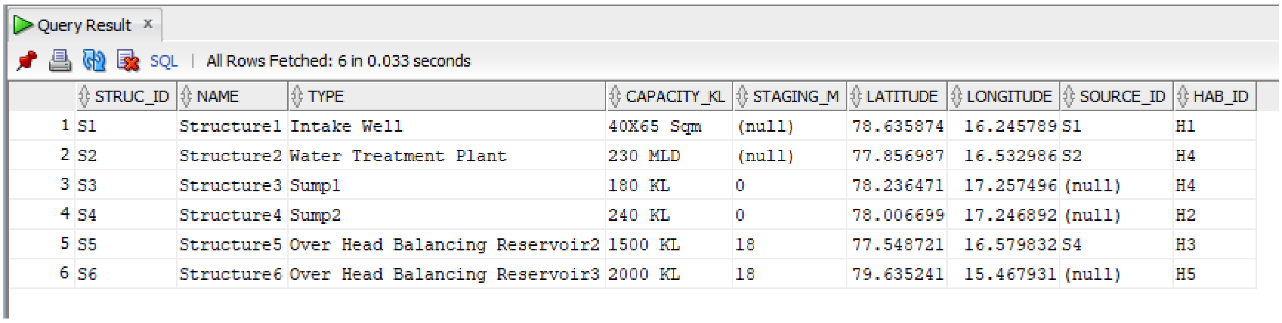
iii) Functional Dependency

Struc_ID→Name, Type, Capacity_KL, Staging_m, Latitude, Longitude,

iv) Sample Data Insertion

```
INSERT INTO "STRUCTURES" (STRUC_ID, NAME, TYPE, CAPACITY_KL, LATITUDE, LONGITUDE, SOURCE_ID, HAB_ID) VALUES ('S1', 'Structure1', 'Intake Well', '40X65 Sqm', '78.635874', '16.245789', 'S1', 'H1')
INSERT INTO "STRUCTURES" (STRUC_ID, NAME, TYPE, CAPACITY_KL, LATITUDE, LONGITUDE, SOURCE_ID, HAB_ID) VALUES ('S2', 'Structure2', 'Water Treatment Plant', '230 MLD', '77.856987', '16.532986', 'S2', 'H4')
INSERT INTO "STRUCTURES" (STRUC_ID, NAME, TYPE, CAPACITY_KL, STAGING_M, LATITUDE, LONGITUDE, HAB_ID) VALUES ('S3', 'Structure3', 'Sump1', '180 KL', '0', '78.236471', '17.257496', 'H4')
INSERT INTO "STRUCTURES" (STRUC_ID, NAME, TYPE, CAPACITY_KL, STAGING_M, LATITUDE, LONGITUDE, HAB_ID) VALUES ('S4', 'Structure4', 'Sump2', '240 KL', '0', '78.006699', '17.246892', 'H2')
INSERT INTO "STRUCTURES" (STRUC_ID, NAME, TYPE, CAPACITY_KL, STAGING_M, LATITUDE, LONGITUDE, SOURCE_ID, HAB_ID) VALUES ('S5', 'Structure5', 'Over Head Balancing Reservoir2', '1500 KL', '18', '77.548721', '16.579832', 'S4', 'H3')
INSERT INTO "STRUCTURES" (STRUC_ID, NAME, TYPE, CAPACITY_KL, STAGING_M, LATITUDE, LONGITUDE, HAB_ID) VALUES ('S6', 'Structure6', 'Over Head Balancing Reservoir3', '2000 KL', '18', '79.635241', '15.467931', 'H5')
```

v) Sample Data Table



All Rows Fetched: 6 in 0.033 seconds

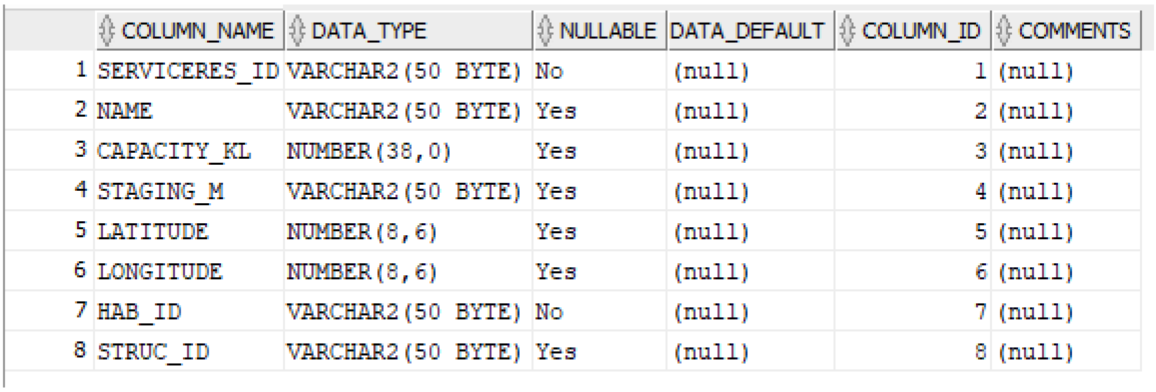
	STRUC_ID	NAME	TYPE	CAPACITY_KL	STAGING_M	LATITUDE	LONGITUDE	SOURCE_ID	HAB_ID
1	S1	Structure1	Intake Well	40X65 Sqm	(null)	78.635874	16.245789	S1	H1
2	S2	Structure2	Water Treatment Plant	230 MLD	(null)	77.856987	16.532986	S2	H4
3	S3	Structure3	Sump1	180 KL	0	78.236471	17.257496	(null)	H4
4	S4	Structure4	Sump2	240 KL	0	78.006699	17.246892	(null)	H2
5	S5	Structure5	Over Head Balancing Reservoir2	1500 KL	18	77.548721	16.579832	S4	H3
6	S6	Structure6	Over Head Balancing Reservoir3	2000 KL	18	79.635241	15.467931	(null)	H5

f. Service Reservoirs Table

i) Create Statement

```
CREATE TABLE servicereservoirs (
    serviceres_id       VARCHAR2(50) NOT NULL,
    name                VARCHAR2(50),
    capacity_kl         INTEGER,
    staging_m           VARCHAR2(50),
    latitude            NUMBER(8, 6),
    longitude           NUMBER(8, 6),
    hab_id              VARCHAR2(50) NOT NULL,
    struc_id            VARCHAR2(50)
);
```

ii) Table Result



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SERVICERES_ID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	NAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	CAPACITY_KL	NUMBER(38,0)	Yes	(null)	3	(null)
4	STAGING_M	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	LATITUDE	NUMBER(8,6)	Yes	(null)	5	(null)
6	LONGITUDE	NUMBER(8,6)	Yes	(null)	6	(null)
7	HAB_ID	VARCHAR2(50 BYTE)	No	(null)	7	(null)
8	STRUC_ID	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)

```
ALTER TABLE servicereservoirs ADD CONSTRAINT servicereservoirs_pk PRIMARY KEY ( serviceres_id );
ALTER TABLE servicereservoirs
    ADD CONSTRAINT serres_hab_fk FOREIGN KEY ( hab_id )
        REFERENCES habitations ( hab_id );

ALTER TABLE servicereservoirs
    ADD CONSTRAINT serres_struc_fk FOREIGN KEY ( struc_id )
        REFERENCES structures ( struc_id );
```

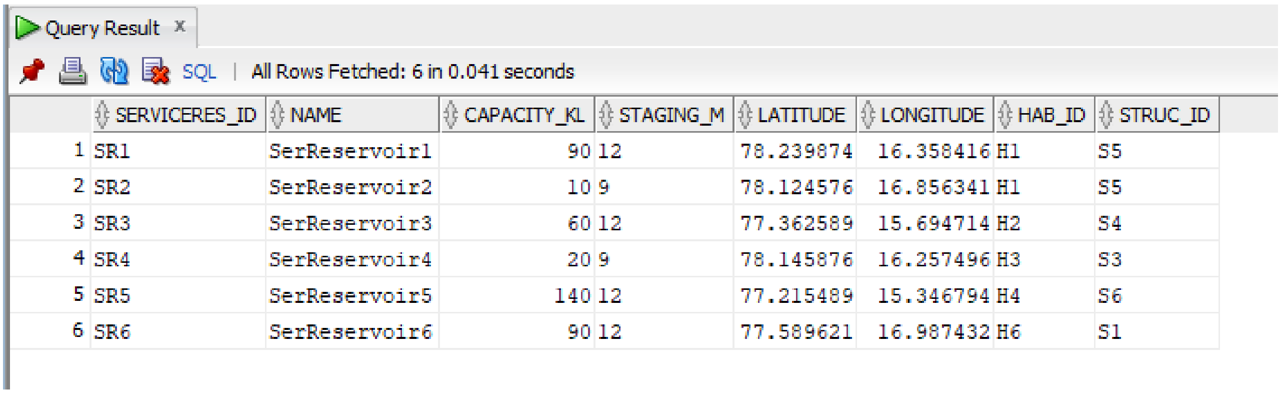
iii) Functional Dependency

ServiceRes_ID→Name, Capacity_KL, Staging_m, Latitude, Longitude

iv) Sample Data Insertion

```
INSERT INTO "SERVICERESERVOIRS" (SERVICERES_ID, NAME, CAPACITY_KL, STAGING_M, HAB_ID, STRUC_ID) VALUES ('SR1', 'SerReservoir1', '90', '12', 'H1', 'S5')
Insert into "SERVICERESERVOIRS" (LATITUDE, LONGITUDE) Values ('78.239874', '16.358416')
INSERT INTO "SERVICERESERVOIRS" (SERVICERES_ID, NAME, CAPACITY_KL, STAGING_M, LATITUDE, LONGITUDE, HAB_ID, STRUC_ID) VALUES ('SR2', 'SerReservoir2', '10', '9', '78.124576', '16.856341', 'H1', 'S5')
INSERT INTO "SERVICERESERVOIRS" (SERVICERES_ID, NAME, CAPACITY_KL, STAGING_M, LATITUDE, LONGITUDE, HAB_ID, STRUC_ID) VALUES ('SR3', 'SerReservoir3', '60', '12', '77.362589', '15.694714', 'H2', 'S4')
INSERT INTO "SERVICERESERVOIRS" (SERVICERES_ID, NAME, CAPACITY_KL, STAGING_M, LATITUDE, LONGITUDE, HAB_ID, STRUC_ID) VALUES ('SR4', 'SerReservoir4', '20', '9', '78.145876', '16.257496', 'H3', 'S3')
INSERT INTO "SERVICERESERVOIRS" (SERVICERES_ID, NAME, CAPACITY_KL, STAGING_M, LATITUDE, LONGITUDE, HAB_ID, STRUC_ID) VALUES ('SR5', 'SerReservoir5', '140', '12', '77.215489', '15.346794', 'H4', 'S6')
INSERT INTO "SERVICERESERVOIRS" (SERVICERES_ID, NAME, CAPACITY_KL, STAGING_M, LATITUDE, LONGITUDE, HAB_ID, STRUC_ID) VALUES ('SR6', 'SerReservoir6', '90', '12', '77.589621', '16.987432', 'H6', 'S1')
```

v) Sample Data Table



All Rows Fetched: 6 in 0.041 seconds

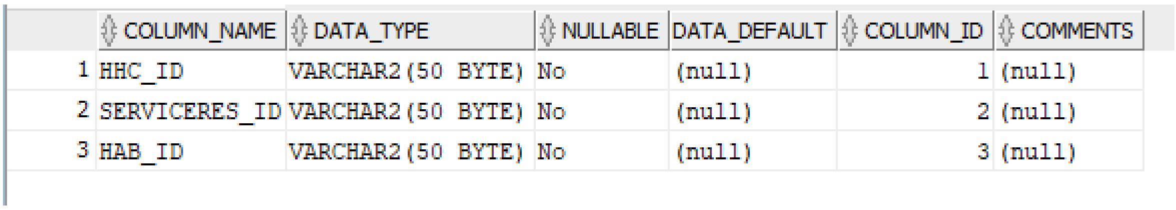
	SERVICERES_ID	NAME	CAPACITY_KL	STAGING_M	LATITUDE	LONGITUDE	HAB_ID	STRUC_ID
1	SR1	SerReservoir1	90	12	78.239874	16.358416	H1	S5
2	SR2	SerReservoir2	10	9	78.124576	16.856341	H1	S5
3	SR3	SerReservoir3	60	12	77.362589	15.694714	H2	S4
4	SR4	SerReservoir4	20	9	78.145876	16.257496	H3	S3
5	SR5	SerReservoir5	140	12	77.215489	15.346794	H4	S6
6	SR6	SerReservoir6	90	12	77.589621	16.987432	H6	S1

g. Household Connections Table

i) Create Statement

```
CREATE TABLE householdconnections (
    hhc_id              VARCHAR2(50) NOT NULL,
    serviceres_id       VARCHAR2(50) NOT NULL,
    hab_id              VARCHAR2(50) NOT NULL
);
```

ii) Table result



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	HHC_ID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	SERVICERES_ID	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	HAB_ID	VARCHAR2(50 BYTE)	No	(null)	3	(null)

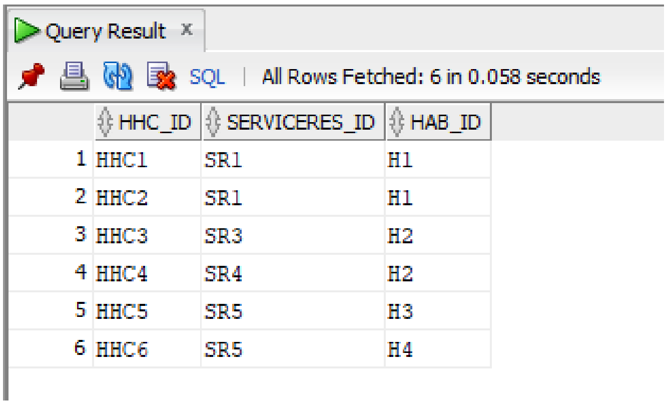
```
ALTER TABLE householdconnections ADD CONSTRAINT householdconnections_pk
PRIMARY KEY ( hhc_id );
ALTER TABLE householdconnections
    ADD CONSTRAINT hhc_hab_fk FOREIGN KEY ( hab_id )
        REFERENCES habitations ( hab_id );

ALTER TABLE householdconnections
    ADD CONSTRAINT hhc_serres_fk FOREIGN KEY ( serviceres_id )
        REFERENCES servicereservoirs ( serviceres_id );
```

iii) Sample Data Insertion

```
INSERT INTO "HOUSEHOLDCONNECTIONS" (HHC_ID, SERVICERES_ID, HAB_ID)
VALUES ('HHC1', 'SR1', 'H1')
INSERT INTO "HOUSEHOLDCONNECTIONS" (HHC_ID, SERVICERES_ID, HAB_ID)
VALUES ('HHC2', 'SR1', 'H1')
INSERT INTO "HOUSEHOLDCONNECTIONS" (HHC_ID, SERVICERES_ID, HAB_ID)
VALUES ('HHC3', 'SR3', 'H2')
INSERT INTO "HOUSEHOLDCONNECTIONS" (HHC_ID, SERVICERES_ID, HAB_ID)
VALUES ('HHC4', 'SR4', 'H2')
INSERT INTO "HOUSEHOLDCONNECTIONS" (HHC_ID, SERVICERES_ID, HAB_ID)
VALUES ('HHC5', 'SR5', 'H3')
```

```
INSERT INTO "HOUSEHOLDCONNECTIONS" (HHC_ID, SERVICERES_ID, HAB_ID)
VALUES ('HHC6', 'SR5', 'H4')
```

iv) Sample Data Table



All Rows Fetched: 6 in 0.058 seconds

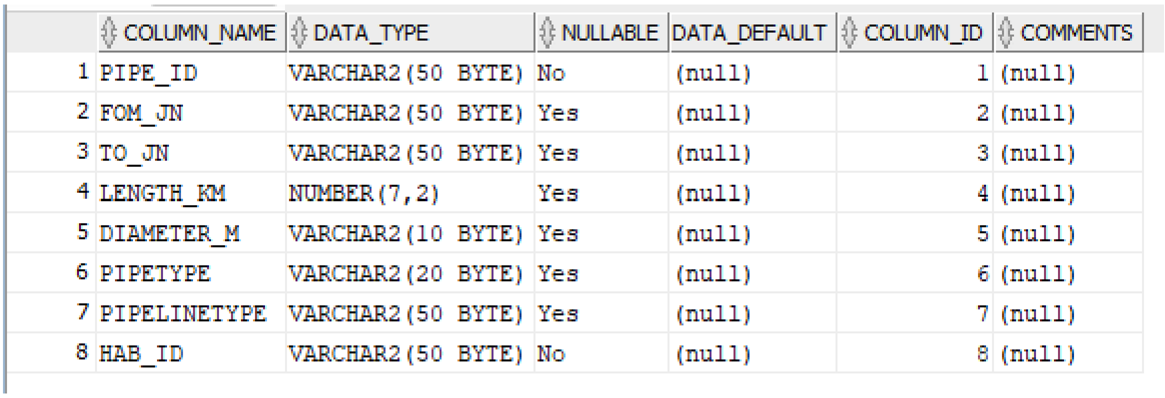
	HHC_ID	SERVICERES_ID	HAB_ID
1	HHC1	SR1	H1
2	HHC2	SR1	H1
3	HHC3	SR3	H2
4	HHC4	SR4	H2
5	HHC5	SR5	H3
6	HHC6	SR5	H4

h. Network Table

i) Create Statement

```
CREATE TABLE network (
    pipe_id      VARCHAR2(50) NOT NULL,
    fom_jn       VARCHAR2(50),
    to_jn        VARCHAR2(50),
    length_km    NUMBER(5, 2),
    diameter_m   VARCHAR2(10),
    pipetype     VARCHAR2(20),
    pipelinetype VARCHAR2(50),
    hab_id       VARCHAR2(50) NOT NULL
);
```

ii) Table Result



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PIPE_ID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	FOM_JN	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	TO_JN	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	LENGTH_KM	NUMBER(7,2)	Yes	(null)	4	(null)
5	DIAMETER_M	VARCHAR2(10 BYTE)	Yes	(null)	5	(null)
6	PIPETYPE	VARCHAR2(20 BYTE)	Yes	(null)	6	(null)
7	PIPELINETYPE	VARCHAR2(50 BYTE)	Yes	(null)	7	(null)
8	HAB_ID	VARCHAR2(50 BYTE)	No	(null)	8	(null)

```
ALTER TABLE network ADD CONSTRAINT network_pk PRIMARY KEY ( pipe_id );
ALTER TABLE network
    ADD CONSTRAINT network_hab_fk FOREIGN KEY ( hab_id )
        REFERENCES habitations ( hab_id );
```

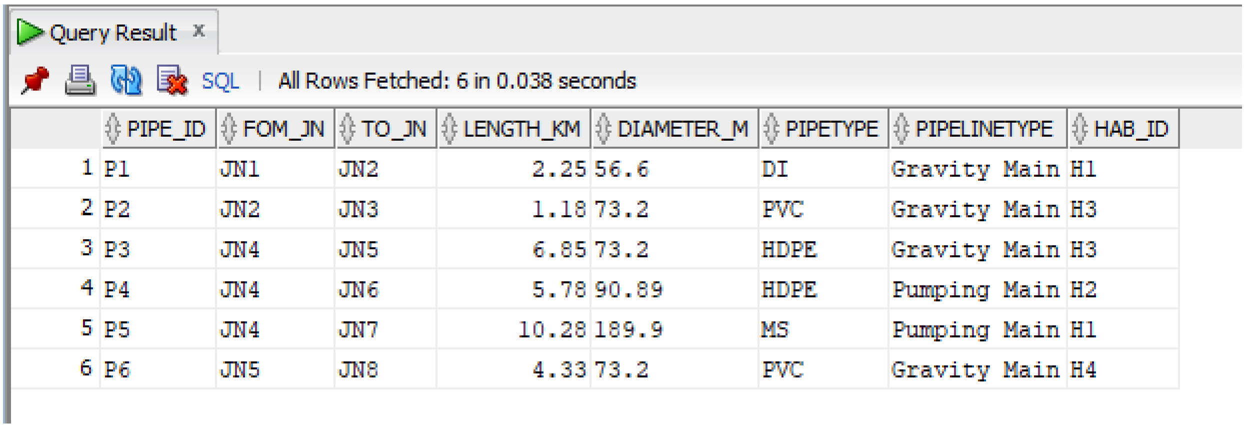
iii) Functional Dependency

Pipe_ID→From_JN, To_JN, Length_KM, Diameter_m, PipeType, PipeLineType

iv) Sample Data Insertion

```
INSERT INTO "NETWORK" (PIPE_ID, FOM_JN, TO_JN, LENGTH_KM, DIAMETER_M, PIPETYPE, PIPELINETYPE, HAB_ID) VALUES ('P1', 'JN1', 'JN2', '2.25', '56.6', 'DI', 'Gravity Main', 'H1')
INSERT INTO "NETWORK" (PIPE_ID, FOM_JN, TO_JN, LENGTH_KM, DIAMETER_M, PIPETYPE, PIPELINETYPE, HAB_ID) VALUES ('P2', 'JN2', 'JN3', '1.18', '73.2', 'PVC', 'Gravity Main', 'H3')
INSERT INTO "NETWORK" (PIPE_ID, FOM_JN, TO_JN, LENGTH_KM, DIAMETER_M, PIPETYPE, PIPELINETYPE, HAB_ID) VALUES ('P3', 'JN4', 'JN5', '6.85', '73.2', 'HDPE', 'Gravity Main', 'H3')
INSERT INTO "NETWORK" (PIPE_ID, FOM_JN, TO_JN, LENGTH_KM, DIAMETER_M, PIPETYPE, PIPELINETYPE, HAB_ID) VALUES ('P4', 'JN4', 'JN6', '5.78', '90.89', 'HDPE', 'Pumping Main', 'H2')
INSERT INTO "NETWORK" (PIPE_ID, FOM_JN, TO_JN, LENGTH_KM, DIAMETER_M, PIPETYPE, PIPELINETYPE, HAB_ID) VALUES ('P5', 'JN4', 'JN7', '10.28', '189.9', 'MS', 'Pumping Main', 'H1')
INSERT INTO "NETWORK" (PIPE_ID, FOM_JN, TO_JN, LENGTH_KM, DIAMETER_M, PIPETYPE, PIPELINETYPE, HAB_ID) VALUES ('P6', 'JN5', 'JN8', '4.33', '73.2', 'pvc', 'Gravity Main', 'H4')
```

v) Sample Data Table

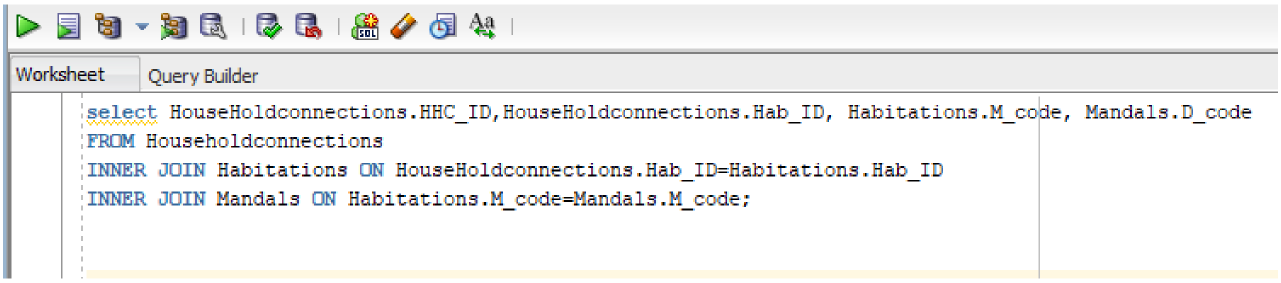


All Rows Fetched: 6 in 0.038 seconds

	PIPE_ID	FOM_JN	TO_JN	LENGTH_KM	DIAMETER_M	PIPETYPE	PIPELINETYPE	HAB_ID
1	P1	JN1	JN2	2.25	56.6	DI	Gravity Main	H1
2	P2	JN2	JN3	1.18	73.2	PVC	Gravity Main	H3
3	P3	JN4	JN5	6.85	73.2	HDPE	Gravity Main	H3
4	P4	JN4	JN6	5.78	90.89	HDPE	Pumping Main	H2
5	P5	JN4	JN7	10.28	189.9	MS	Pumping Main	H1
6	P6	JN5	JN8	4.33	73.2	PVC	Gravity Main	H4

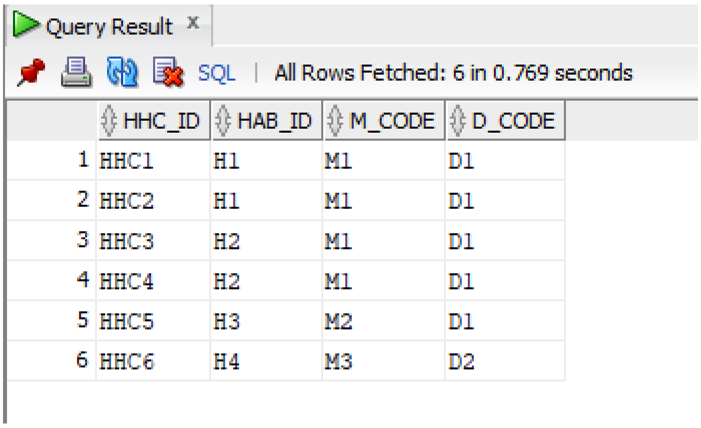
6. Queries and Output

Output to query 1: To get household connection location in a state



```
select HouseHoldconnections.HHC_ID,HouseHoldconnections.Hab_ID, Habitations.M_code, Mandals.D_code
FROM Householdconnections
INNER JOIN Habitations ON HouseHoldconnections.Hab_ID=Habitations.Hab_ID
INNER JOIN Mandals ON Habitations.M_code=Mandals.M_code;
```

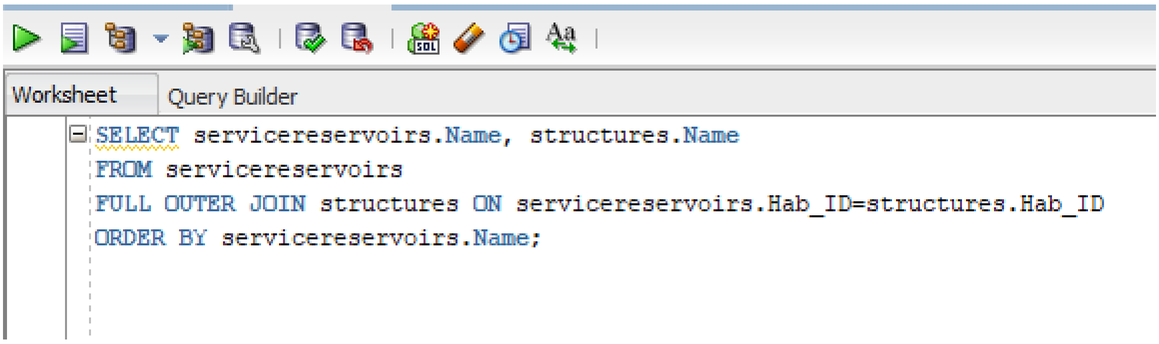
And the result is



All Rows Fetched: 6 in 0.769 seconds

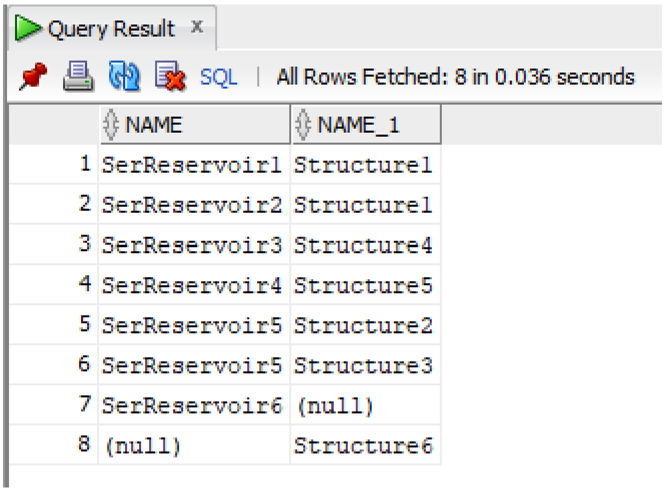
	HHC_ID	HAB_ID	M_CODE	D_CODE
1	HHC1	H1	M1	D1
2	HHC2	H1	M1	D1
3	HHC3	H2	M1	D1
4	HHC4	H2	M1	D1
5	HHC5	H3	M2	D1
6	HHC6	H4	M3	D2

Output to query 2: To get which structures are watered by the sources



```
SELECT servicereservoirs.Name, structures.Name
FROM servicereservoirs
FULL OUTER JOIN structures ON servicereservoirs.Hab_ID=structures.Hab_ID
ORDER BY servicereservoirs.Name;
```

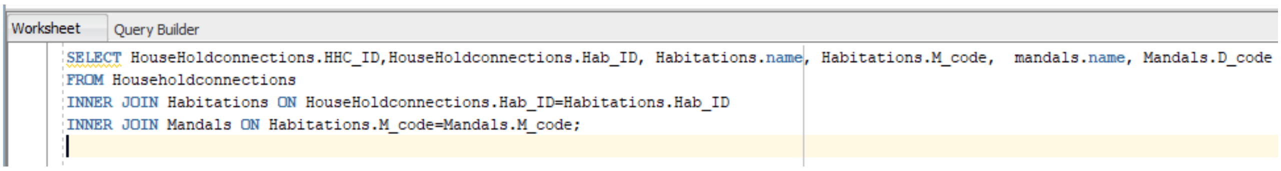
And the result is



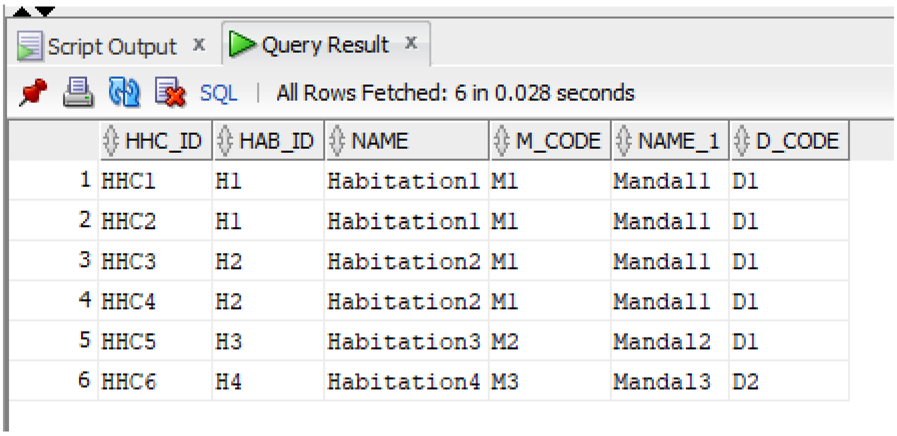
All Rows Fetched: 8 in 0.036 seconds

	NAME	NAME_1
1	SerReservoir1	Structure1
2	SerReservoir2	Structure1
3	SerReservoir3	Structure4
4	SerReservoir4	Structure5
5	SerReservoir5	Structure2
6	SerReservoir5	Structure3
7	SerReservoir6	(null)
8	(null)	Structure6

Output to query 3: To get habitation and Mandal IDs & Names along with District IDs for household connections.



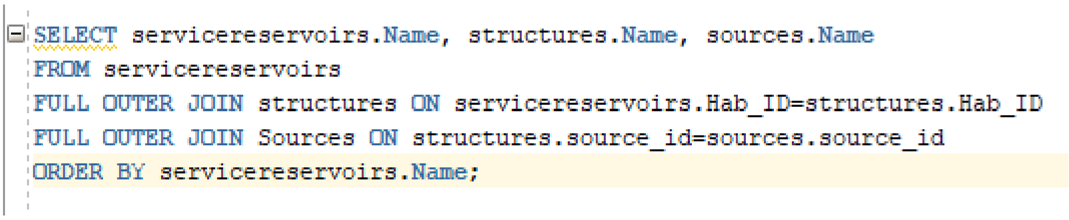
```
SELECT HouseHoldconnections.HHC_ID,HouseHoldconnections.Hab_ID, Habitations.name, Habitations.M_code,  mandals.name, Mandals.D_code
FROM Householdconnections
INNER JOIN Habitations ON HouseHoldconnections.Hab_ID=Habitations.Hab_ID
INNER JOIN Mandals ON Habitations.M_code=Mandals.M_code;
```



All Rows Fetched: 6 in 0.028 seconds

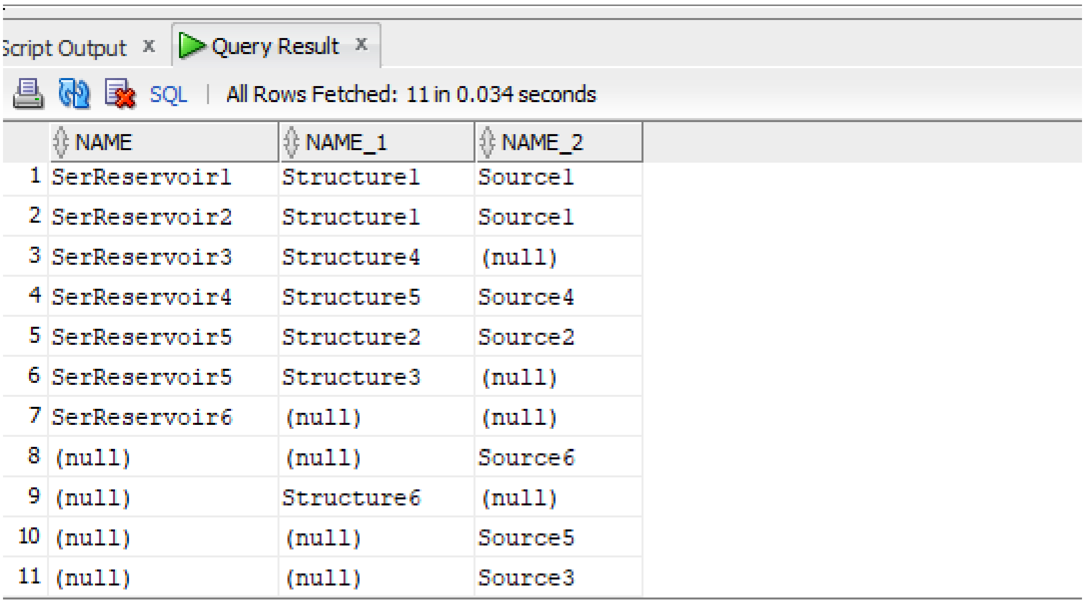
	HHC_ID	HAB_ID	NAME	M_CODE	NAME_1	D_CODE
1	HHC1	H1	Habitation1	M1	Mandal1	D1
2	HHC2	H1	Habitation1	M1	Mandal1	D1
3	HHC3	H2	Habitation2	M1	Mandal1	D1
4	HHC4	H2	Habitation2	M1	Mandal1	D1
5	HHC5	H3	Habitation3	M2	Mandal2	D1
6	HHC6	H4	Habitation4	M3	Mandal3	D2

Output to query 4: To find out the connections of all structures and sources with service reservoirs



```
SELECT servicereservoirs.Name, structures.Name, sources.Name
FROM servicereservoirs
FULL OUTER JOIN structures ON servicereservoirs.Hab_ID=structures.Hab_ID
FULL OUTER JOIN Sources ON structures.source_id=sources.source_id
ORDER BY servicereservoirs.Name;
```

And the result is



	NAME	NAME_1	NAME_2
1	SerReservoir1	Structure1	Source1
2	SerReservoir2	Structure1	Source1
3	SerReservoir3	Structure4	(null)
4	SerReservoir4	Structure5	Source4
5	SerReservoir5	Structure2	Source2
6	SerReservoir5	Structure3	(null)
7	SerReservoir6	(null)	(null)
8	(null)	(null)	Source6
9	(null)	Structure6	(null)
10	(null)	(null)	Source5
11	(null)	(null)	Source3

7. IMPLEMENTATION NOTES

November 10th, 2021 – Decided the topic of the project and worked on rough ER Diagram

November 18th, 2021 – Created Sources, District, Mandals and Habitations Tables in SQL Data Modeler

November 22nd, 2021 – Worked on creating structures, service reservoirs tables

November 25th, 2021 – Designed ER Diagram for Water Distribution System Management in SQL Data Modeler

December 4th, 2021 – Concluded with ER Diagram with cardinalities and catalogued all required data into Excel Spreadsheet

December 7th, 2021 – Imported DDL statements to Oracle SQL Developer

December 10th, 2021 – Started inserting data into the tables which are already created

December 12th, 2021 – Sample data insertion was done

December 15th, 2021– Created SQL queries and ran them in SQL Data Developer

December 16th, 2021 – Worked on documentation and prepared for presenting the project in class

December 17th, 2021 – Worked on future enhancements and known problems and finalized the report to be submitted

8. KNOWN PROBLEMS

1. This database is currently designed for limited pipeline networks. It can be upgraded or modified to accommodate different types of pipelines like gravity main and pumping main, one in each pipeline network table.
2. Segment data field in this database is duplicated in two tables. But this can be eliminated either by inserting more fields in the tables or by creating new cardinalities.
3. In real time, every structure should have a habitation connected to it but in query 2, there are few unconnected structures.

9. FUTURE ENHANCEMENTS

1. The network table can be upgraded to accommodate little other type of pipelines too.
2. By segregating all the structures, service reservoirs and household connections with latitude, latitude and by denoting them with a district, mandal and habitation code, query can be drawn to extract a structure by its geographical location.
3. In pipe line network, sometimes the from_JN can be one habitation and to_JN can be in another. This can be found extended the network table to from_JNHab and to_JNHab, and relate this table to habitations table.
4. The database can be extended to water refilling and water billing systems.
5. The household connections table can be extended upgrade customer information.
6. Also the database must be upgraded in a timely manner patches may be needed to enhance the quality of output.