|  |
| --- |
|  |
| Database design for modeling a Oil-Well |
|  |

|  |
| --- |
| Contents  [Abstract 2](#_Toc70608584)  [Mission statement 2](#_Toc70608585)  [Mission objectives 2](#_Toc70608586)  [E/R diagram 3](#_Toc70608587)  [ER Diagram (normalized) 4](#_Toc70608588)  [Relational model after normalization 5](#_Toc70608589)  [A complete list of use-cases 8](#_Toc70608590)  [Use case realization. 9](#_Toc70608591)  [Insert 9](#_Toc70608592)  [Delete 9](#_Toc70608593)  [Update 9](#_Toc70608594)  [Aggregate Query 9](#_Toc70608595)  [Join Query 9](#_Toc70608596)  [Test plan and records. 10](#_Toc70608597)  [Conclusion 12](#_Toc70608598)  [Reference 12](#_Toc70608599) |

# Abstract

A reservoir simulator’s main goal is to accurately predict Well’s flow through the tubing from perforations to the Wellhead and to the downstream. To represent accurately through a database is quite paramount to representing such a complex model; a bad representation can do a lot of damage in time and monetarily. The data-model is used to demonstrate wells are connected to various facilities. As more wells are added, it become easier to manage them in a relational database vs file system, where changing a data requires updating all the files etc.

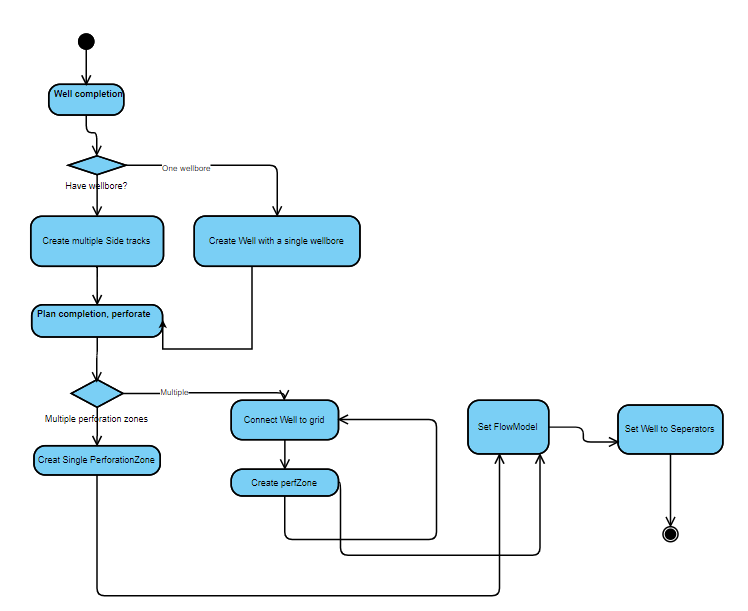
# Mission statement

To represent a Well and its relations in a relational database.

# Mission objectives

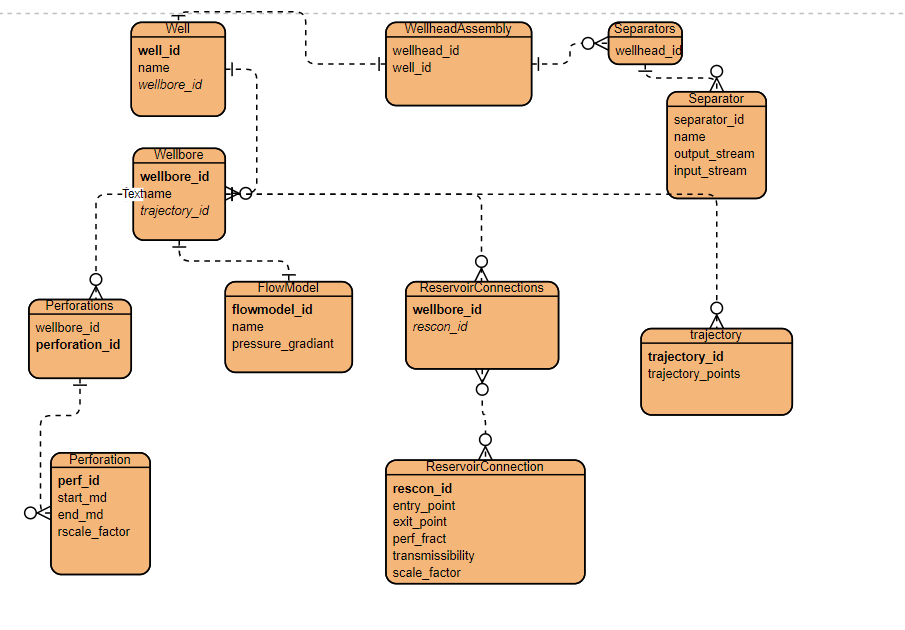
The use case for this activity is completion of a well. This represents only activities at a high level, and what is needed for reservoir simulation. Drilling a well in itself is a complex operation. Completion of a well on the other hand is about making the oil flow through it and bring it to the Wellhead and connect it to the downstream facilities like separators for separation into oil, gas, water.

Following is the activity diagram for well completion; starting by planning a well completion, followed by wellbores or sidetracks. A wellbore is drilled per its defined trajectory and angles at which it can be bent; it is controlled by XYZ or points on the casing’s axis. Once the wellbores have been drilled and cemented in place, the next step is to complete the well, by perforating some length of casing and tubing through which the oil and other mixes flow into the well. A flow model is defined to represent how oil is going to flow through the well and if it will have enough pressure to naturally flow to the top due to pressure gradient in the tube. The well or wellhead is connected to the facilities on platform for separation in oil, gas, water.

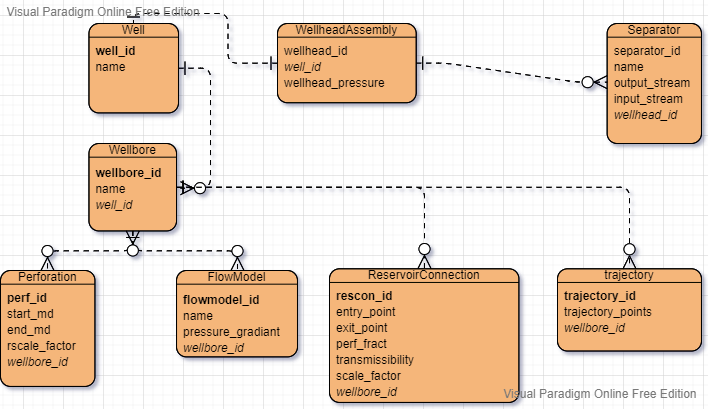


# E/R diagram

An E/R diagram for your proposed database. Underline key attributes for entity sets, specify relationships with arrowheads.



# ER Diagram (normalized)



# Relational model after normalization

For each table, show the attribute dependencies, and explain why it is in 1NF, 2NF, or 3NF;

Well(**well\_id**, name)

For relation Well, it’s represented by two attributes *well\_id*, and name; **well\_id** is the primary key. This is in 3NF.

Wellbore(**wellbore\_id**, name, well\_id)

For relation Wellbore, it’s represented by three attributes *wellbore\_id*, name, and well\_id. Wellbore\_id is the primary key, and well\_id is the foreign key. This is normalized to 3NF after well\_id is added as a foreign key.

The original table was in not normalized cause well and wellbore were both in the same table. Since the well\_id would have to be repeated for multiple wellbores, i.e., it’s a one-to-many relation, it’s normalized to 3NF after creating a separate table for wellbore.

Perforation (perforation\_id, start\_md, end\_md, rscale\_factor, wellbore\_id)

Perforation is defined by the section of the pipe towards end of the well. The holes are created in using penetrating guns for the oil to flow from it towards the wellhead.

A perforation has perforation\_id as its primary key and has a start measured depth, end measured depth, rscale\_factor, and a wellbore\_id. Wellbore\_id act as its foreign key since this has one to many relationship with a wellbore.

FlowModel(flow\_model\_id, name, pressure\_gradient)

Flow model defines how the oil will flow in the tubing and based on the measured distance between wellhead and perforation, a pressure drop/gradient is calculated. This will help in calculating if the oil will be able to automatically flow from bottom of the well to the wellhead without any need for artificial lift.

Each wellbore has a flow model and multiple wellbores can have many flowmodels; similarly a flow model can have many wellbores, so it’s a many to many relationship.

It’s defined by flow\_model\_id, name, pressure\_gradient, and wellbore\_id. Flowmodel\_id is it’s primary\_key and wellbore\_id is it’s foreign key.

FlowModelWellboreMap(**flow\_model\_wellbore\_id**, flow\_model\_id, wellbore\_id);

Flowmodelwellboremap maps the flowmodel with a wellbore. Since the relation between a flow model and wellbore is many to many, it’s natural to associate a flow model to be used in multiple wellbores. Simillarly, multiple wellbore can use any of the flow model, which substantiate the many to many relationship.

ReservoirConnection (reservoir\_connection\_id, entry\_md, entry\_point, exit\_md, exit\_point, rscale\_factor, perf\_frac, transmissibility, scale\_factor, wellbore\_id)

Reservoir connections are the virtual flow controls or the holes through which the oil is going to flow. These can be multiple ones per wellbore, and the relation is defined as with following attributes

reservoir\_connection\_id, entry\_md, entry\_point, exit\_md, exit\_point, rscale\_factor, perf\_frac, transmissibility, scale\_factor, wellbore\_id

reservoir\_connection\_is acts as its primary\_key and wellbore\_id as its foreign key.

Separator(**separator\_id**, name, output\_stream, input\_stream)

The flow stream from wellhead (top of the well) contains a mix of oil, gas and water, which are fed to a separator(s), which separate them into multiple streams by iteratively separating them in multiple separators.

separator\_id, name, output\_stream, input\_stream

separator\_id is its primary key. Name attribute represents a name for a separator; input\_stream represents the content of the stream going into a separator, and output\_stream is the multiple streams that the content will be broken into. Multiple wellheads are connected to multiple separators, so the relation is many to many.

create database uhd5318;

use uhd5318;

create database uhd5318;

use uhd5318;

CREATE TABLE Well (

well\_id int NOT NULL AUTO\_INCREMENT,

well\_name NVARCHAR(30) NOT NULL,

PRIMARY KEY (well\_id)

);

CREATE TABLE Wellbore (

wellbore\_id int NOT NULL AUTO\_INCREMENT,

wellbore\_name NVARCHAR(30),

well\_id int NOT NULL,

PRIMARY KEY (wellbore\_id),

FOREIGN KEY (well\_id) REFERENCES Well(well\_id)

);

CREATE TABLE Perforation(

perforation\_id int NOT NULL AUTO\_INCREMENT,

start\_md DOUBLE NOT NULL,

end\_md DOUBLE NOT NULL,

rscale\_factor DOUBLE NOT NULL,

wellbore\_id INT NOT NULL,

PRIMARY KEY (perforation\_id),

FOREIGN KEY (wellbore\_id) REFERENCES Wellbore(wellbore\_id)

);

CREATE TABLE FlowModel(

flow\_model\_id int NOT NULL AUTO\_INCREMENT,

name nvarchar(50) NOT NULL,

pressure\_gradient DOUBLE NOT NULL,

PRIMARY KEY (flow\_model\_id)

);

CREATE TABLE FlowModelWellboreMap(

flow\_model\_wellbore\_id int NOT NULL AUTO\_INCREMENT,

flow\_model\_id int NOT NULL,

wellbore\_id INT NOT NULL,

PRIMARY KEY (flow\_model\_wellbore\_id),

FOREIGN KEY (flow\_model\_id) REFERENCES FlowModel(flow\_model\_id),

FOREIGN KEY (wellbore\_id) REFERENCES Wellbore(wellbore\_id)

);

Create Table ReservoirConnection (

reservoir\_connection\_id int NOT NULL AUTO\_INCREMENT,

entry\_md double NOT NULL,

entry\_point\_x double NOT NULL,

entry\_point\_y double NOT NULL,

entry\_point\_z double NOT NULL,

exit\_md double NOT NULL,

exit\_point\_x double NOT NULL,

exit\_point\_y double NOT NULL,

exit\_point\_z double NOT NULL,

rscale\_factor double NOT NULL,

perf\_frac double NOT NULL,

transmissibility double NOT NULL,

scale\_factor double NOT NULL,

wellbore\_id INT NOT NULL,

PRIMARY KEY (reservoir\_connection\_id),

FOREIGN KEY (wellbore\_id) REFERENCES Wellbore(wellbore\_id)

);

Create Table Wellhead(

wellhead\_id int NOT NULL AUTO\_INCREMENT,

well\_id INT NOT NULL,

PRIMARY KEY (wellhead\_id),

FOREIGN KEY (well\_id) REFERENCES Well(well\_id)

);

Create Table SeparatorFacility(

separator\_id int NOT NULL AUTO\_INCREMENT,

name NVARCHAR(30) NOT NULL,

output\_stream NVARCHAR(30) NOT NULL,

input\_stream NVARCHAR(30) NOT NULL,

primary key(separator\_id)

);

Create Table SeparatorFacilityWellMap(

separatorfacility\_well\_id int NOT NULL AUTO\_INCREMENT,

separator\_id int NOT NULL,

wellhead\_id INT NOT NULL,

primary key(separatorfacility\_well\_id),

FOREIGN KEY (separator\_id)

REFERENCES SeparatorFacility(separator\_id),

FOREIGN KEY (wellhead\_id)

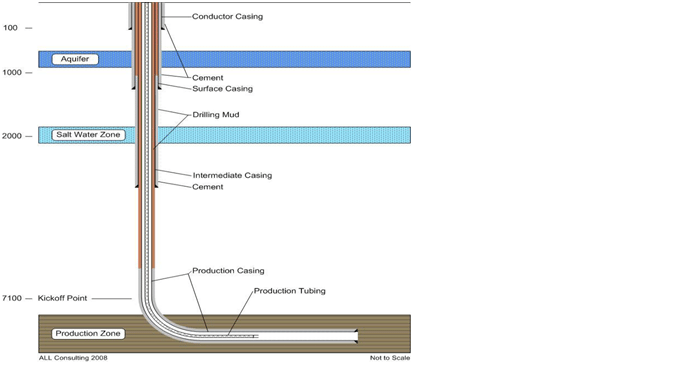
REFERENCES wellhead(wellhead\_id)

);

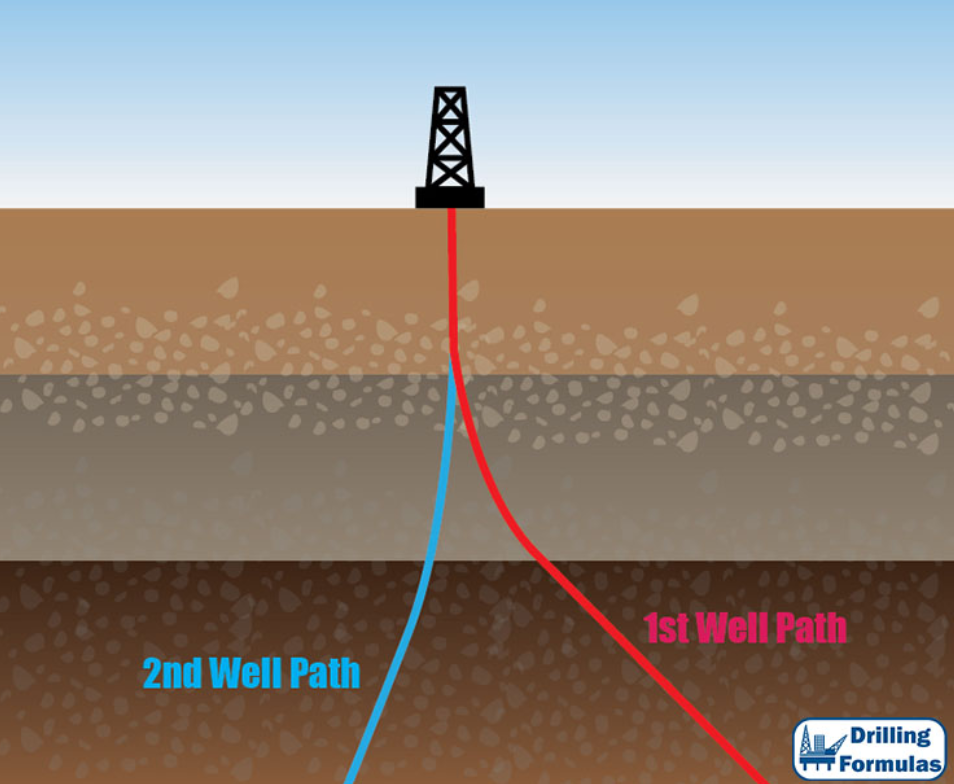
# A complete list of use-cases

The use case for this activity is completion of a well. This represents only activities at a high level, and what’s needed for reservoir simulation. Drilling a well in itself is a complex operation. Completion of a well on the other hand is about making the oil flow through it and bring it to the Wellhead and connect it to the downstream facilities like separators for separation into oil, gas, water.

Well



Two wellbores



[1-Multiple-exploration-wells-from-a-single-wellbore.jpg (750×750) (drillingformulas.com)](http://www.drillingformulas.com/wp-content/uploads/2017/02/1-Multiple-exploration-wells-from-a-single-wellbore.jpg)

Following is the activity diagram for well completion; starting by planning a well completion, followed by wellbores or sidetracks. A wellbore is drilled per its defined trajectory and angles at which it can be bent; it is controlled by XYZ or points on the casing’s axis.

Once the wellbores has been drilled and cemented in place, the next step is to complete the well, by perforating some length of casing and tubing through which the oil and other mixes flow into the well. A flow model is defined to represent how oil is going to flow through the well and if it will have enough pressure to naturally flow to the top due to pressure gradient in the tube. The well or wellhead is connected to the facilities on platform for separation in oil, gas, water.

# Use case realization.

(include the SQL statements that implement every use case). Once again, be sure to have: for each entity, at least one insert, one delete, one update, and one aggregate query; and for each set of entities that have a direct relationship, one joint query.

## Insert

INSERT INTO well(well\_name) values("P1");

INSERT INTO wellbore(wellbore\_name, well\_id) values("P1", 1);

INSERT INTO PERFORATION(start\_md, end\_md,rscale\_factor, wellbore\_id) VALUES(11000, 11500, .1, 1);

INSERT INTO FlowModel(name, pressure\_gradient) VALUES('const-pressure-0', 0.89892);

INSERT INTO FlowModelWellboreMap(flow\_model\_id, wellbore\_id) values(1, 1);

INSERT INTO ReservoirConnection(entry\_md,entry\_point\_x,entry\_point\_y,

entry\_point\_z, exit\_md, exit\_point\_x, exit\_point\_y, exit\_point\_z, rscale\_factor, perf\_frac, transmissibility, scale\_factor, wellbore\_id) VALUES(11000, 1128, 1891, 11000, 11500, 1128, 1891, 11500, 0.8, .8, 23.23, .5, 1);

INSERT INTO Wellhead(well\_id) VALUES(1);

INSERT INTO SeparatorFacility(name,output\_stream,input\_stream) VALUES("sep-owg", "ow,g", "owg");

INSERT INTO SeparatorFacilityWellMap(wellhead\_id,separator\_id) VALUES(1,1);

## Delete

Delete from wellhead where well\_id = 1;

Delete from reservoirconnection where well\_id = 1 and start\_md>10000;

## Update

## Aggregate Query

select count(reservoir\_connection\_id), reservoirconnection.wellbore\_id from reservoirconnection, wellbore where reservoirconnection.wellbore\_id = wellbore.wellbore\_id group by wellbore.wellbore\_id;

## Join Query

# Test plan and records.

For every use case, specify the input data, the expected output, and the actual output. Correct any bugs you find.

1. Show all wells

MariaDB [uhd5318]> select \* from well;

+---------+-----------+

| well\_id | well\_name |

+---------+-----------+

| 1 | P1 |

| 2 | P2 |

| 3 | P3 |

| 4 | P4 |

| 5 | P5 |

| 6 | I1 |

| 7 | I2 |

| 8 | I3 |

| 9 | I4 |

| 10 | I5 |

+---------+-----------+

10 rows in set (0.001 sec)

1. Show count of wellbores for each well

MariaDB [uhd5318]> select count(wellbore\_id), well\_id from wellbore group by well\_id;

+--------------------+---------+

| count(wellbore\_id) | well\_id |

+--------------------+---------+

| 2 | 1 |

| 2 | 2 |

| 2 | 3 |

| 2 | 4 |

| 2 | 5 |

| 1 | 6 |

| 1 | 7 |

| 1 | 8 |

| 1 | 9 |

| 1 | 10 |

+--------------------+---------+

10 rows in set (0.001 sec)

1. Show number of perforations for each wellbore

MariaDB [uhd5318]> select count(perforation\_id), perforation.wellbore\_id from perforation, wellbore where perforation.wellbore\_id = wellbore.wellbore\_id group by wellbore.wellbore\_id;

+-----------------------+-------------+

| count(perforation\_id) | wellbore\_id |

+-----------------------+-------------+

| 1 | 1 |

| 1 | 2 |

| 1 | 3 |

| 1 | 4 |

| 1 | 5 |

| 1 | 6 |

| 1 | 7 |

| 1 | 8 |

| 1 | 9 |

| 1 | 10 |

| 2 | 11 |

| 1 | 12 |

| 1 | 13 |

| 1 | 14 |

| 1 | 15 |

+-----------------------+-------------+

15 rows in set (0.001 sec)

1. Show number of reservoir connections for each wellbore

MariaDB [uhd5318]> select count(reservoir\_connection\_id), reservoirconnection.wellbore\_id from reservoirconnection, wellbore where reservoirconnection.wellbore\_id = wellbore.wellbore\_id group by wellbore.wellbore\_id;

+--------------------------------+-------------+

| count(reservoir\_connection\_id) | wellbore\_id |

+--------------------------------+-------------+

| 1 | 1 |

| 1 | 2 |

| 1 | 3 |

| 1 | 4 |

| 1 | 5 |

| 1 | 6 |

| 1 | 7 |

| 1 | 8 |

| 1 | 9 |

| 1 | 10 |

| 1 | 11 |

| 1 | 12 |

| 1 | 13 |

| 1 | 14 |

| 1 | 15 |

+--------------------------------+-------------+

15 rows in set (0.001 sec)

1. Show that the reservoir connections are within the perforation’s interval.
2. Show how many times a flow model is used in each well.
3. Show what’s being produced by each well and the sequence of produce from each separator level
4. Delete a well, should delete related entities
5. Insert a new well and show it’s flow model and separators.
6. Delete a wellbore and should delete related values for dependent relation.

# Conclusion

# Reference

1. [erdiagram-normalized.vpd - VP Online (visual-paradigm.com)](https://online.visual-paradigm.com/w/dmqvjtwy/diagrams/#diagram:proj=0)
2. <https://mariadb.com/kb/>
3. [well-perforation-tubing.png (683×384) (enverus.com)](https://www.enverus.com/wp-content/uploads/2014/10/well-perforation-tubing.png)
4. [1-Multiple-exploration-wells-from-a-single-wellbore.jpg (750×750) (drillingformulas.com)](http://www.drillingformulas.com/wp-content/uploads/2017/02/1-Multiple-exploration-wells-from-a-single-wellbore.jpg)