## Description

We are going to split up into groups and build a database in AWS MySQL.

As part of a follow-up, you will think about the concepts for a database to be loaded into AWS. You will need to deploy a VPC with private and public subnet options, stand-up a simple EC2 instance and install both a database using the dimensional version of the OLTP database provided to you.

Here are the learning objectives of the lab:

1. Learn how to develop a simple dimensional model from defined requirements
2. Develop a dimensional database from an ERD
3. Understand the basics of AWS environments including VPCs, subnets, EC2, Redshift

This lab is split into two parts 1) the first part is work developed in class; and 2) take home exercise to develop and finish on your own.

Here are the steps that this lab will take:

*Part I – in class*

* Step 1: Set-up and deploy OLTP database in AWS (20 pts)
* Step 2: Inspect the Database and Update with a foreign key constraint (10 pts)
* Step 3: Develop analytic requirements (20 pts)

*Part II – at home*

* Step 4: Define conceptual and physical dimensional (facts and dimensions) model (50 pts)

# Part I: Build the Environment

## Step 1: Set-up and deploy OLTP database in AWS (20 pts)

AWS provides cloud-based database relational database services (RDS). Amazon RDS can be used to deploy a MySQL instance in the cloud. In this step, we will prepare our cloud infrastructure to be able to build the AWS Instance in. With your group, decide on which instance of AWS you will use, and deploy the following database (dav6100\_db\_2) to the cloud.

You can use the resources on Canvas to assist with running the script to create a database in the cloud.

A close up of a map

Description automatically generated

## Step 2. Inspect the Database and add the Foreign Key Constraint (10 pts)

Run some scripts to assist in inspecting the database (profiling). Scripts like counts of records, sums of totals, analysis of keys and indices help you understand the size of the database you are going to transform and what records might be relevant.

You will note that t\_ord\_delivery table does not have a home, it is left out! Let’s add a foreign key constraint for status\_code and ord\_id to enhance the relationships across the relevant tables.

Run the **dav6100\_db\_2\_foreign\_keys.sql** script.

Step 3: Develop analytic requirements (20 pts)

*3.1 Develop the analytic requirements, brainstorming activity*

**Scenario:** The service line manager at your organization would like to have quick, ready access to the procurement and contracting data that would be available to them and their staff. They would like to understand how many invoices and how many payments they are making per month and what the average turnaround time is from the time they receive and invoice to the time they make payments. Data in the source database (dav6100\_db\_2) can be considered transactional and contains purchase order information, invoicing, procurement, vendor and contracting data. Refer to Kimball (Ch.5) Dimensional Modeling for Procurement use case as reference.

**Activity:** As a class, we will brainstorm some Facts that the service line manager. Record the results in a dimensional modeling template. I am your business team. Here are the four steps to designing the dimensional model: 1) Select the business process (procurement); 2) Declare the grain; 3) Identify the dimensions; 4) Identify the Facts.

Here are some of the questions that may inform the information you brainstorm: [[1]](#footnote-1)

* What type of routine analysis do you currently perform? What data is used? How do you currently get the data? What do you do with the information once you get it?
* What analysis would you like to perform? Are there potential improvements to your current method/process?
* What type of on-the-fly, ad hoc analysis do you typically perform?
* What do they do with the analysis? Do you have time to ask the follow-up questions?
* Which reports do you currently use? What data on the report is important? How do you use the information? If the report were dynamic, what would the report do differently?
* How much historical information is required?

**Questions**:

1. What is the grain of the data that we will look for?
2. What are the dimensions? What type of dimensions are there?
3. What are the facts?

Please provide your responses in the space provided below and then insert an excel spreadsheet with the data warehouse bus matrix.

*3.2 Define conceptual and logical data model to inform design*

Now that you have identified the dimensions and facts important to your service line manager, you need to begin translating this into a model that would be suited for an analytic database. Here are the steps we will follow[[2]](#footnote-2):

1. High-level model with granularity. A good place to start is to create a conceptual data model. It can be a simple bubble chart or an ERD with simple
2. Detailed design with table-by-table attributes.[[3]](#footnote-3) You should include use surrogate keys, define primary and foreign keys and identify logical relationships through expression of cardinality.
3. Validate the deign with IT and Business stakeholders
4. Finalize the design

# Part II: Develop the Dimensional Model

Since you defined the grain in Part 1, use the high-level conceptual framework approach and develop the simple conceptual model for a Star Schema below.

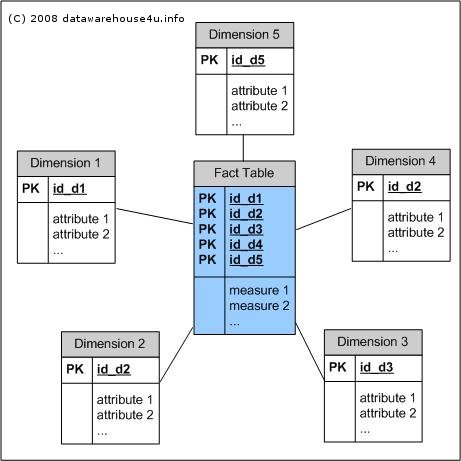
## Step 4: Detailed Dimensional Model (50 pts)

*4.1 Define conceptual and logical data model to inform design*

Now that you have identified the dimensions and facts important to your service line manager, you need to begin translating this into a model that would be suited for an analytic database. You can use [DB Diagram](https://dbdiagram.io/d) as an open source tool to develop your model conceptual model. Chapter 5 (Kimball) will assist in framing the dimensional modeling discussion.

Here are the steps we will follow[[4]](#footnote-4):

1. High-level model with granularity. A good place to start is to create a conceptual data model. It can be a simple bubble chart or an ERD.
2. Detailed design with table-by-table attributes.[[5]](#footnote-5) You should include use surrogate keys, define primary and foreign keys and identify logical relationships through expression of cardinality.
3. Validate the deign with IT and Business stakeholders (we will fake this)
4. Finalize the design



*4.2 Develop physical model*

Now that we have drawn out the dimensions, you should develop a detailed dimensional model. Take the following steps to determine the dimensional model.

1. Using the Physical Layout Template [here](https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/books/data-warehouse-dw-lifecycle-toolkit/), record the basic facts and dimensions. Be sure to add information regarding the type of dimension that to be created. Refer to Kimball (2016) to assist with dimensional definitions.



1. After you have laid out your dimensions and facts in a physical diagram, you should start creating the physical data model in MySQL. Write the DDL script to create the facts and dimensions.
2. Export the MySQL DDL file and save it.

## Submission

To submit, respond to the Lab 01 assignment and provide:

1. an image of your conceptual model
2. the physical layout template
3. your reversed engineered MySQL diagram
4. and the DDL file.

Submit the assignment with the following naming conventions in the files:

<<Group\_#>>\_<<name of file>>\_Lab\_01

Example:

Group\_1\_conceptual\_model\_Lab\_01

1. <https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/books/data-warehouse-dw-lifecycle-toolkit/> [↑](#footnote-ref-1)
2. Ch. 18 p. 434 of Kimball (2016). The Definitive Guide to Dimensional Modeling. [↑](#footnote-ref-2)
3. <https://www.kimballgroup.com/1998/05/surrogate-keys/> [↑](#footnote-ref-3)
4. Ch. 18 p. 434 of Kimball (2016). The Definitive Guide to Dimensional Modeling. [↑](#footnote-ref-4)
5. <https://www.kimballgroup.com/1998/05/surrogate-keys/> [↑](#footnote-ref-5)