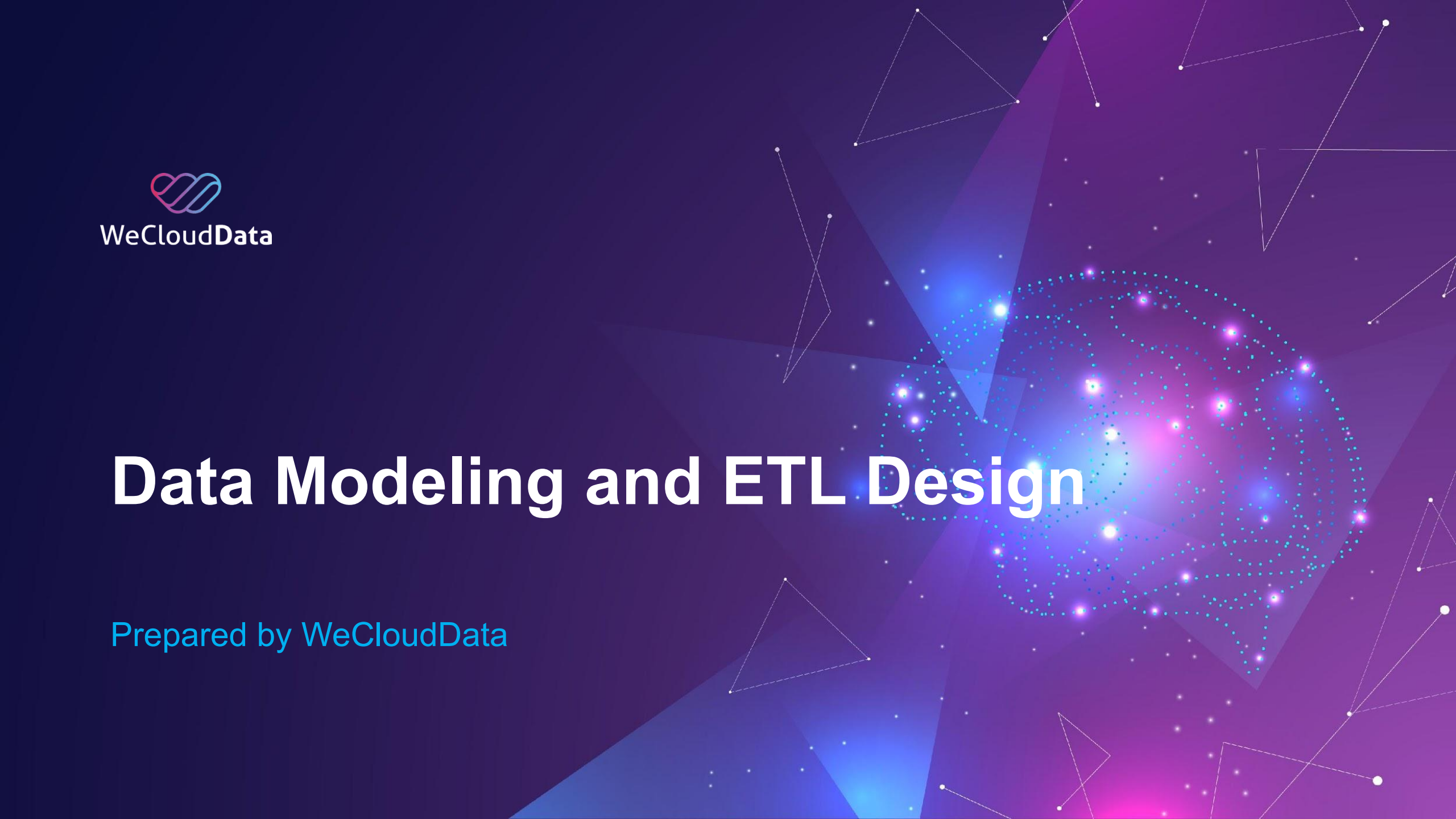


Data Modeling and ETL Design

Prepared by WeCloudData

Data Warehousing Concepts

What is a data warehouse?

- A data warehouse is a type of data management system that is designed to enable and **support business intelligence** (BI) activities, especially **analytics**.
- Data warehouses are solely intended to perform **queries and analysis** and often contain large amounts of historical data.
- The data within a data warehouse is usually derived from a **wide range of sources** such as application log files and transaction applications.

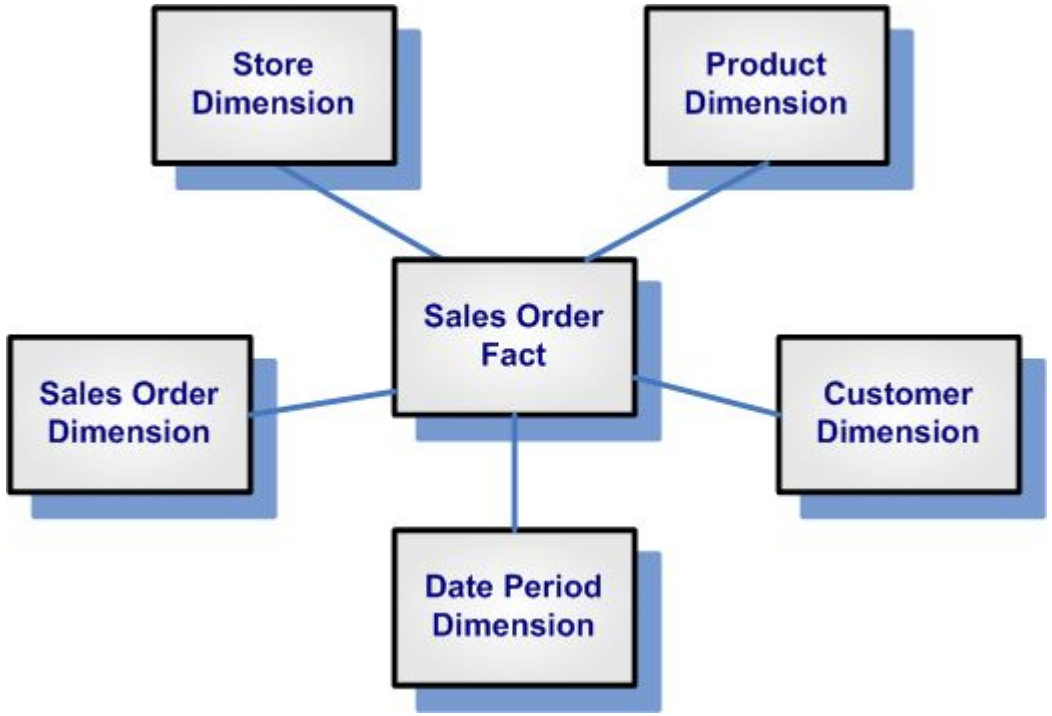
OLTP databases and OLAP databases

- **OLTP** (Online Transaction Processing) databases: Transactions based databases used for **processing transactional data**. The databases like Mysql, MS Sql server, Oracle DB2, belong to this type. It is usually that DBAs are responsible for the management of OLTP.
- **OLAP** (Online Analytical Processing) databases: This type of database that enables users to easily and selectively extract and **query data in order to analyze** it from different points of view.

The Data Modelling Comparison of OLTP and OLAP

	OLTP	OLAP
Examples	MySQL, MS SQL	Snowflake, Redshift, BigQuery
Purposes	Business Transaction	Aggregated Analysis
Primary Interaction	Single Transaction	Aggregated Transaction
Interaction Method	Insert, Update, Delete	Select
Temporal Focus	Current	Current/ Historic
Design Optimization	Update Concurrency	High-performance Queries
Design Principle	3NF (normalization)	Star Schema, Snowflake Schema (denormalization)

Data Model Schema in Data Warehousing



Fact Table (s) + Dimension Tables

- Dimension tables only connect to Fact tables, 2 dimension tables or 2 fact tables will not connect.

- The purpose of the data modelling in data warehouse is the **DA or BI tools can simply query with the simple clause SELECT, WHERE and GROUP BY clauses.** Making sure people can easily build his/her own reports and answer his/her own questions.
- Dimension tables are denormalized (e.g. address in both Store and Customer table)

Fact table

Data Model

- A fact table is found at the **center** of a schema surrounded by dimension tables.
- A fact table consists of facts of a particular business process e.g., sales revenue by month by product.
- Facts are also known as **measurements** or **metrics**. A fact table record captures a measurement or a metric.

Measure types

Additive – additive measures are measures that can be added to all dimensions, such as ***sales amount***.

Non-additive – different from additive measures, non-additive measures are measures that cannot be added to all dimensions, such of ***inventory level of each day***.

Semi-additive – semi-additive measures are the measure that can be added to only some dimensions and not across other, such as ***balance amounts*** which is a common semi-additive fact because they are additive across all dimensions except time.

Dimension table

Data Model

Dimension table:

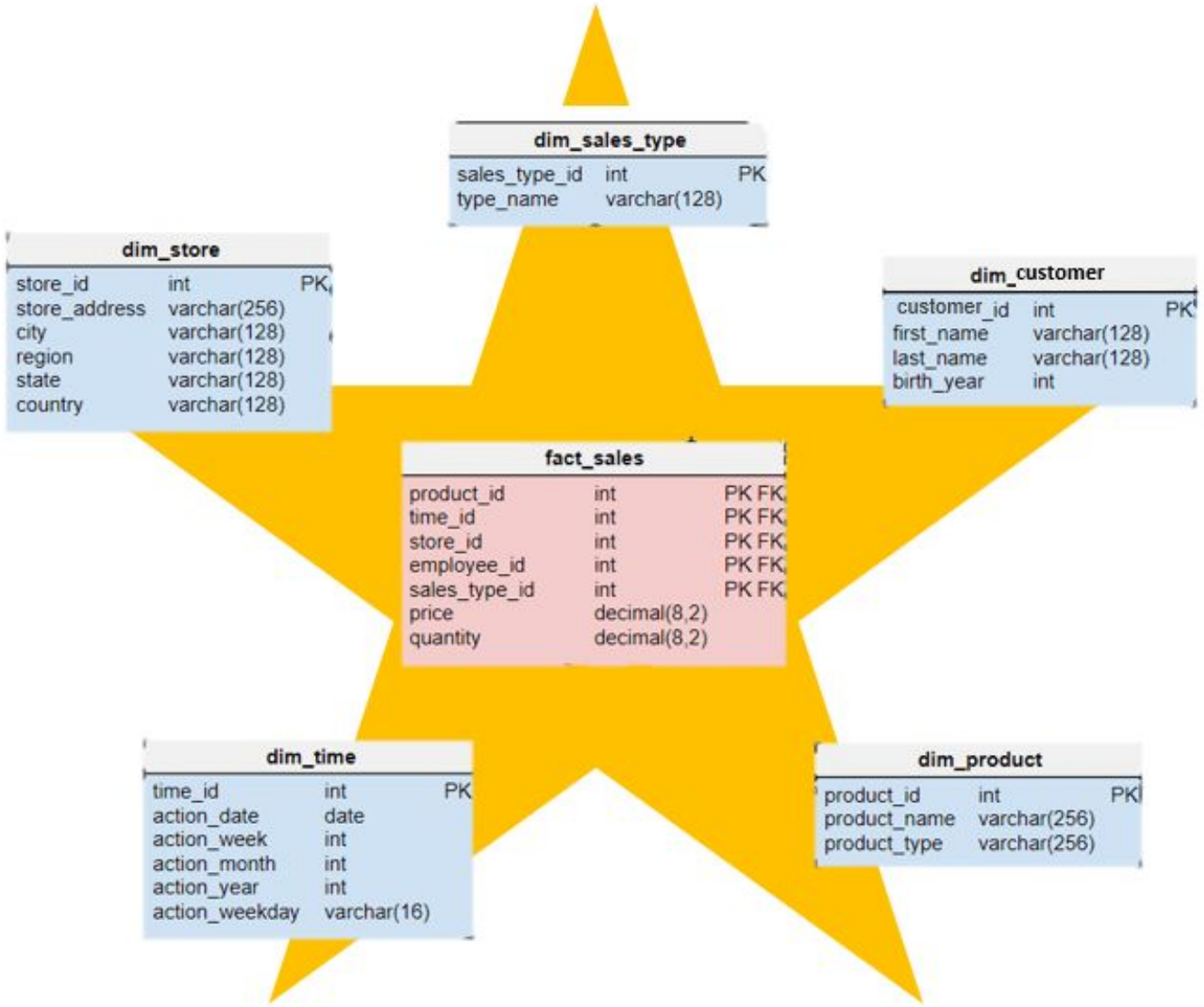
- A dimension table is one of the companion tables to a fact table. A dimension table contains the textual descriptor of the business.
- The fields of the dimension table are designed to satisfy these two important requirements:
  - Query constraining / grouping / filtering.
  - Report labeling.
- Dimension tables are joined to fact table via a key.
- Dimension tables are denormalized tables.
- The dimension can also contain **one or more hierarchical relationships** (Category, brand, product)

Fact and Dimension table Comparison

Data Model

	Fact Table	Dimension Table
Define	Measurements, metrics or facts about a business process.	Companion table to the fact table contains descriptive attributes to be used as query constraining.
Characteristic	at the center of a schema surrounded by dimensions.	Connected to the fact table.
Design	Defined by their grain or its most atomic level.	Should be descriptive, complete, and quality assured.
Task	Fact table is a measurable event for which dimension table data is collected and is used for analysis.	Collection of reference information about a business.
Data	contain quantitative information like sales amount.	Contains attributes described the details of the dimension.
Keys	Foreign keys to Dimensions.	Primary key that uniquely identifies each dimension.
Hierarchy	None	Contains Hierarchies. For example Product could contain brand, category etc.

Data Model Schema – Star Schema

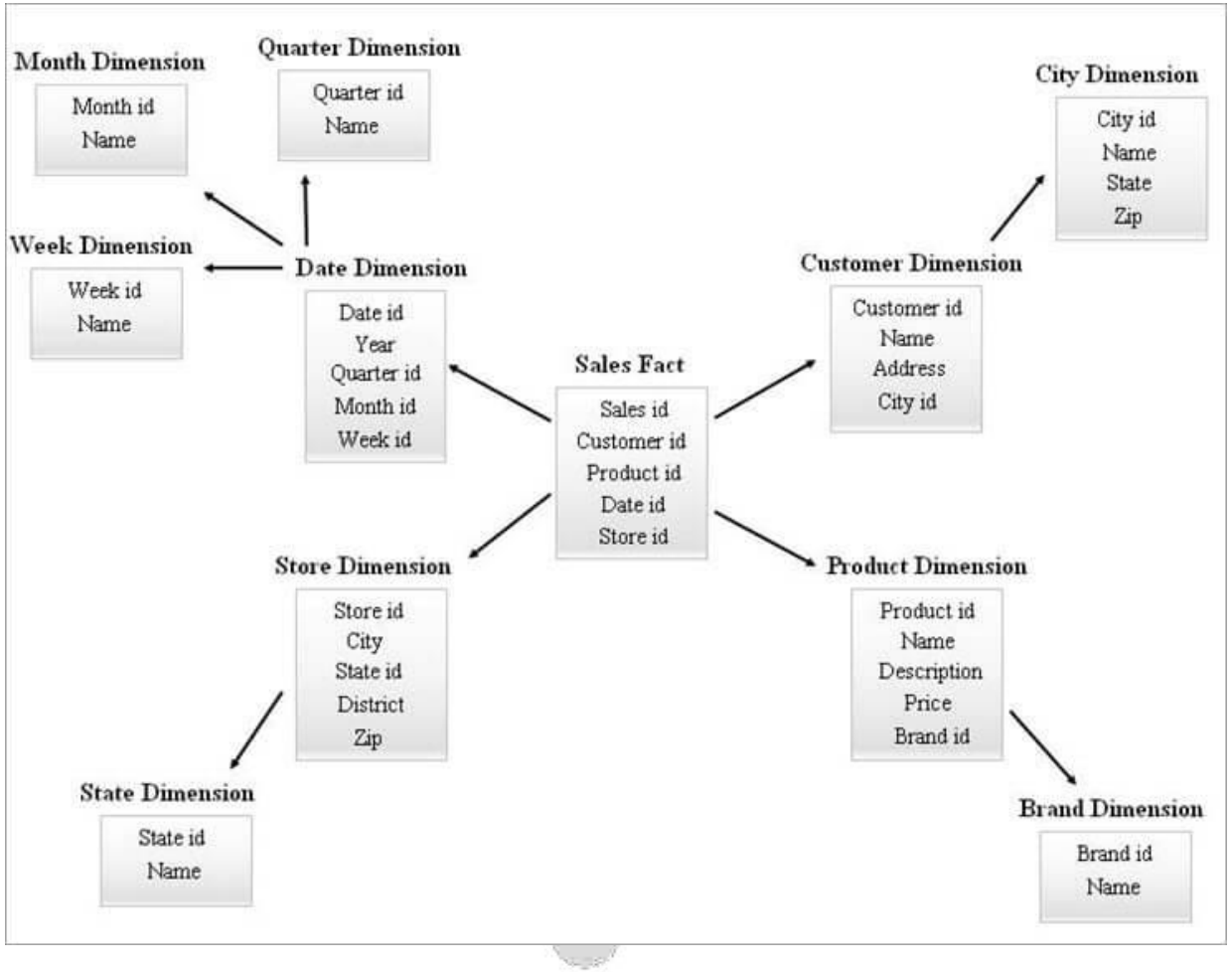


What is Star Schema?
Star Schema use (a) large fact table(s) to store transactional or measured data, and several smaller dimensional tables that store attributes about the data.

All the information in a area will be stored in one dimension table.

How does Star Schema work?
The Fact tables(s) will join the unique dimension table to get the aggregate values insight in the area.

Data Model Schema – Snowflake Schema



What is Snowflake Schema?
The snowflake schema is a variant of the star schema. The difference is that unlike star schema, the dimensions are present in a normalized form in multiple related tables.

How does Star Schema work?
The Fact tables(s) will join the set of tables in the dimension to get the aggregate values insight in the area.

4 Steps to Create a Data Model in a data warehouse

1. **Identify the business requirements;**
   Example:
   "As a marketing manager, I need to know the number of products the customer bought last year in order to target them with an upsell offer."
   From the story above, we can determine that we will need to aggregate the number of products per customer based on sales from the previous year.

1. **Identify the grain in the fact table(s);**
   - To identify what is one row in the fact table represent. Choose the most atomic level (lowest level) for one row.

	EmpKey	DateKey	CustomerKey	ProductKey	SupplierKey	QuantityOrdered	UnitPrice	UnitCost	CostValue	CostSales	Margin
1	5	19960704	85	11	5	12	14.00	21.00	168.00	252.00	84.00
2	5	19960704	85	42	20	10	9.80	14.00	98.00	140.00	42.00
3	5	19960704	85	72	14	5	34.80	34.80	174.00	174.00	0.00

   - Atomic level is the level where data can not be subdivided or split any more. In your BI reports, if you have a table contain data for every day, then the grain is day.
   - The grains of all the dimensions should be considered.

4 Steps to Create a Data Model in a data warehouse

3. **Identify the dimensions;**
   - Dimensions are the attributes like date, store, inventory, etc. These dimensions are where all the data should be stored. For example, the date dimension may contain data like a year, month and weekday.
   - Dimensions may contain hierarchies, such as product dimension may contain: Category, brand, and Series.

3. **Identify the measures in the fact table(s).**
   - Translate the general business requirements into quantitative measures;
   - Identify the columns from the measures.
   - Don’t leave a user(a DA or a BI tool) to calculate measures on Run Time.

Data Model Design – Principle

Data Model

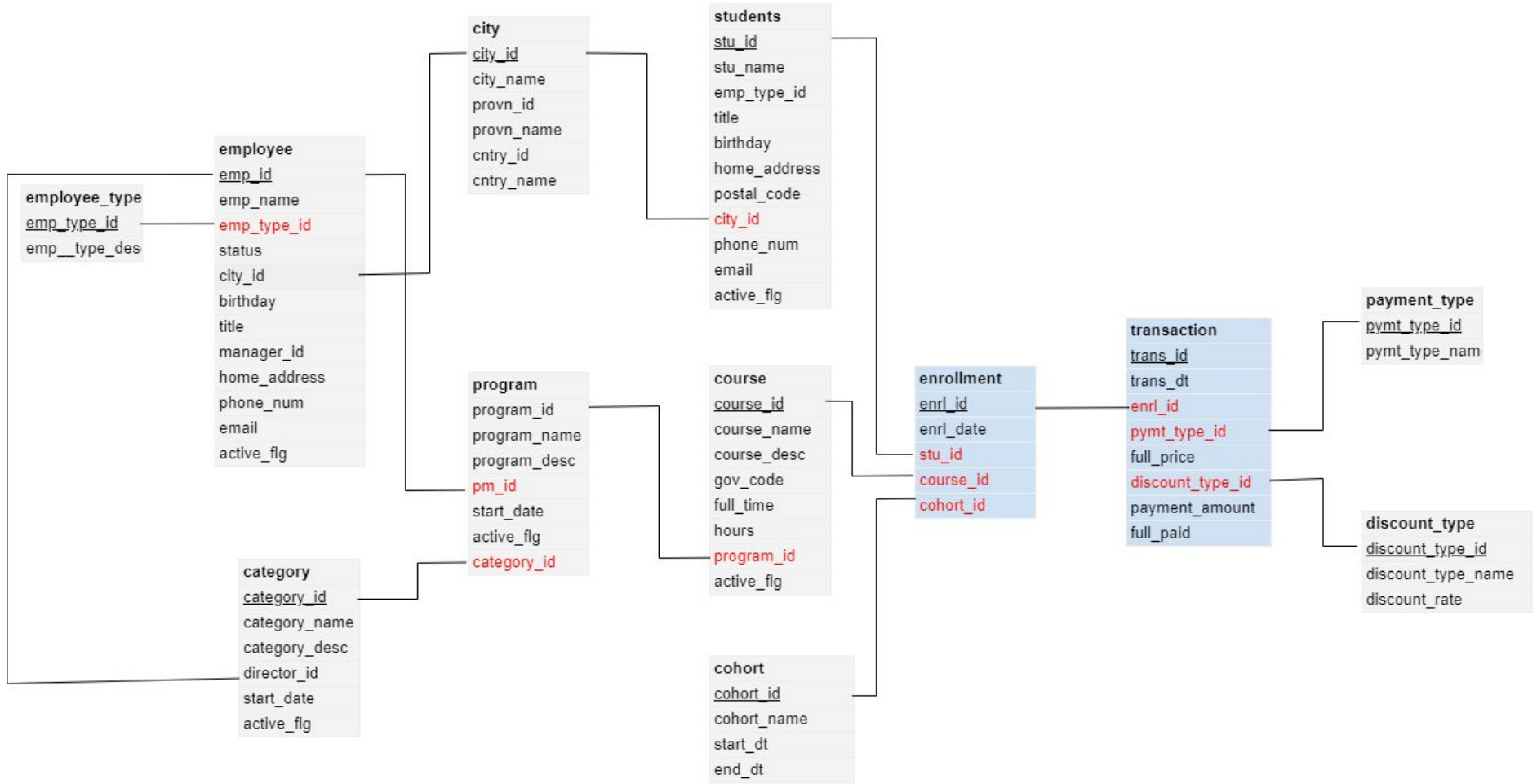
“Even with the absence of a BI tool in place, a person who know basic SQL skill still can develop reports tables with simple queries incorporating SELECT, WHERE and GROUP BY clauses.”

Case Study

A School Database

Data Model

The database in the operating system include the transactions and enrollment tables and also the



Detailed Dataset

WeCloudData

Step 1 – Business Requirements

Data Model

1. **What is the trend of each course enrollment?**
2. **Which course is the most popular course?**
3. **Each course's everyday enrollment?**
4. **The full payment rate of each program in each cohort?**
5. **What is the average discount rate of each course each cohort?**
6. **Which cohorts is over-promoted?**

What if we don't use data warehouse only use queries to query from the original database?

The drawbacks are:

- We need to write complicated queries for each requirement each time;
- If the data size are big (e.g 100G a table), the performance will be very slow;
- It is not a best practice to run queries directly on transaction database, which will slow down the performance of the database.
- This is why we need data warehouse and data modelling for data warehouse.

Step 1 – Business Requirements – *Project Target*

Data Model

Our Target

- We can write complex queries to get the results for the above requirements as a Data Analyst.
- But in the data warehouse project, instead, we need a data model that can:
  - provide a simple way to query, the end users will get results with simple SELECT, JOIN and GROUP BY, etc . **[Simple]**
  - provide a flexible way that can fit the various requirements. **[Flexible]**

General Solution

- Understand the business scope **–** what this project is about? Is about sales, payment or employee?
- **Simple** - Measures ready in the fact table. All the related information is ready in a dimension table(denormalization).
- **Flexible** - Atomic granularity to meet different levels of hierarchies. For example, in the requirements, it is asking each day, we can also query each week and each month if required.

Step 1 – Business Requirements – *Requirements Analytics*

Data Model

1. **The trend of each course enrollment?**
   Translation: *For each course, how many students enroll in each cohort.*
1. **Which course is the most popular course?**
   Translation: On average which course has the most students enrollment?
1. **Everyday enrollment?**
2. *Translation:* Each course everyday enrollment.
3. **The full payment rate of each program in each cohort?**
   Translation: by the starting date of the cohort, what is the full payment rate？
1. **What is the average discount rate of each course each cohort?**
2. **how many cohorts is over-promoted?**
   Translation: For each course in each cohort, if the average discount rate of the cohort is larger than the average discount rate of the entire course, then count as one over-promoted？

(You can think how to write queries for each above question if we query directly from the operational database.)

Step 1 – Business Requirements – Breakdown

Data Model

1. **For each course, how many students enroll in each cohort.**
   sum(students) *on cohort level , on course level (program/ category level)*
1. **On average which course has the most students enrollment?**
   avg(*sum(student)* on cohort /course level) on course level)
1. **Each course everyday enrollment.**
   sum(students) *on day level ((week/month/ year level)), on course level (program/ category level)*
1. **By the starting date of the cohort, what is the full payment rate？**
   count(full_paid) *on cohort /course level*
1. **What is the average discount rate of each course each cohort?**
   avg(discount_rate) *on cohort /course level*
1. **For each course in each cohort, if the average discount rate of the cohort is larger than the average discount rate of the entire course, then count as one over-promoted？**
   sum**(*case when* [*avg(discount_rate)* on cohort /course level] > [*avg(discount_rate)* on course level] then 1)**

Step 2&3 Grain and Dimension

Data Model

Dimensions

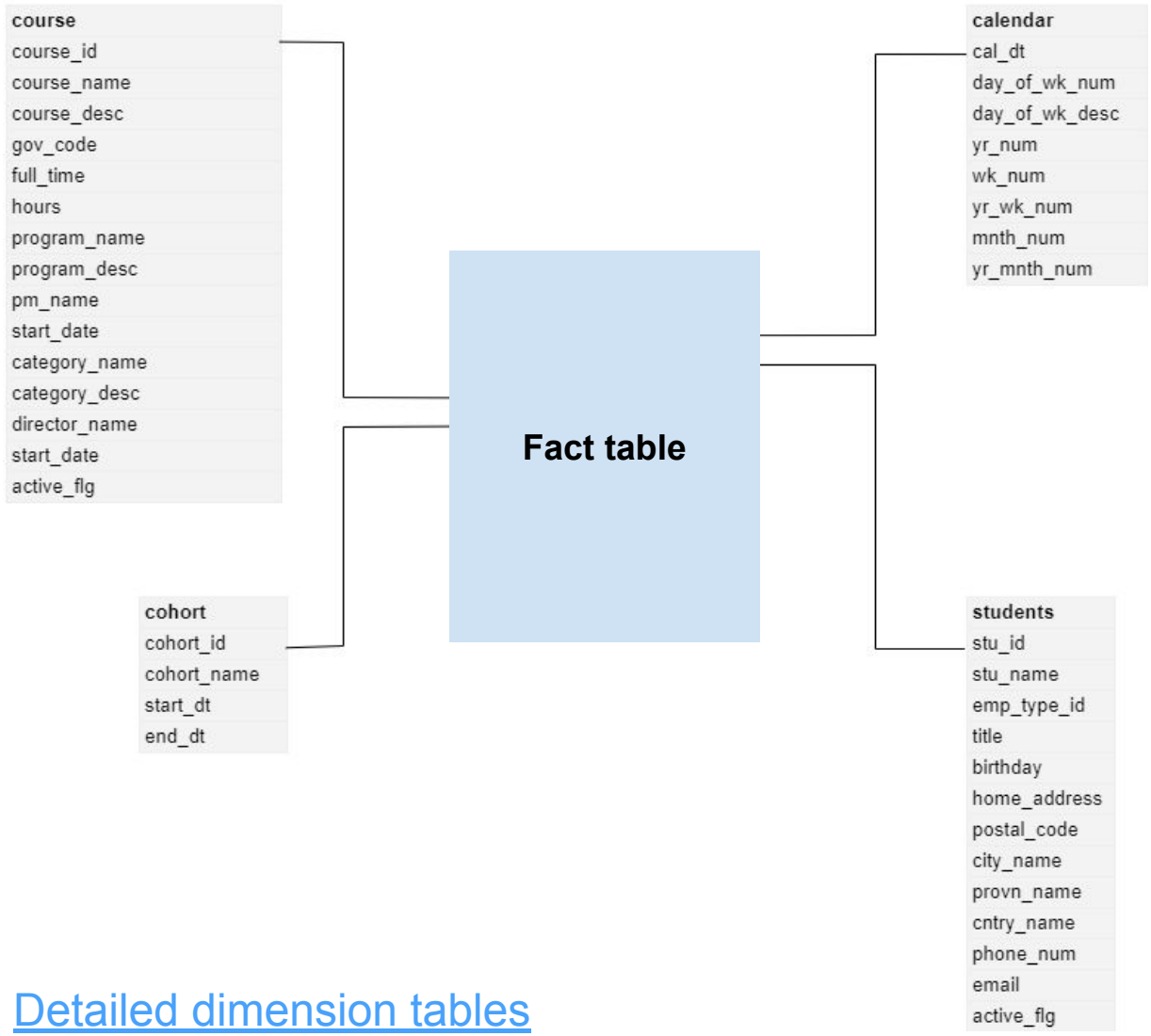
- **Course (Program/Category)**
- **Cohort**
- **Date (week/ Month / Year)**
- **Student**
- *! Employee*
- *~ Payment type*
- *~ Discount type*

Grain

- course
- cohort
- date
- student

One atomic row= *date + course_id + cohort_id + student_id + measures*

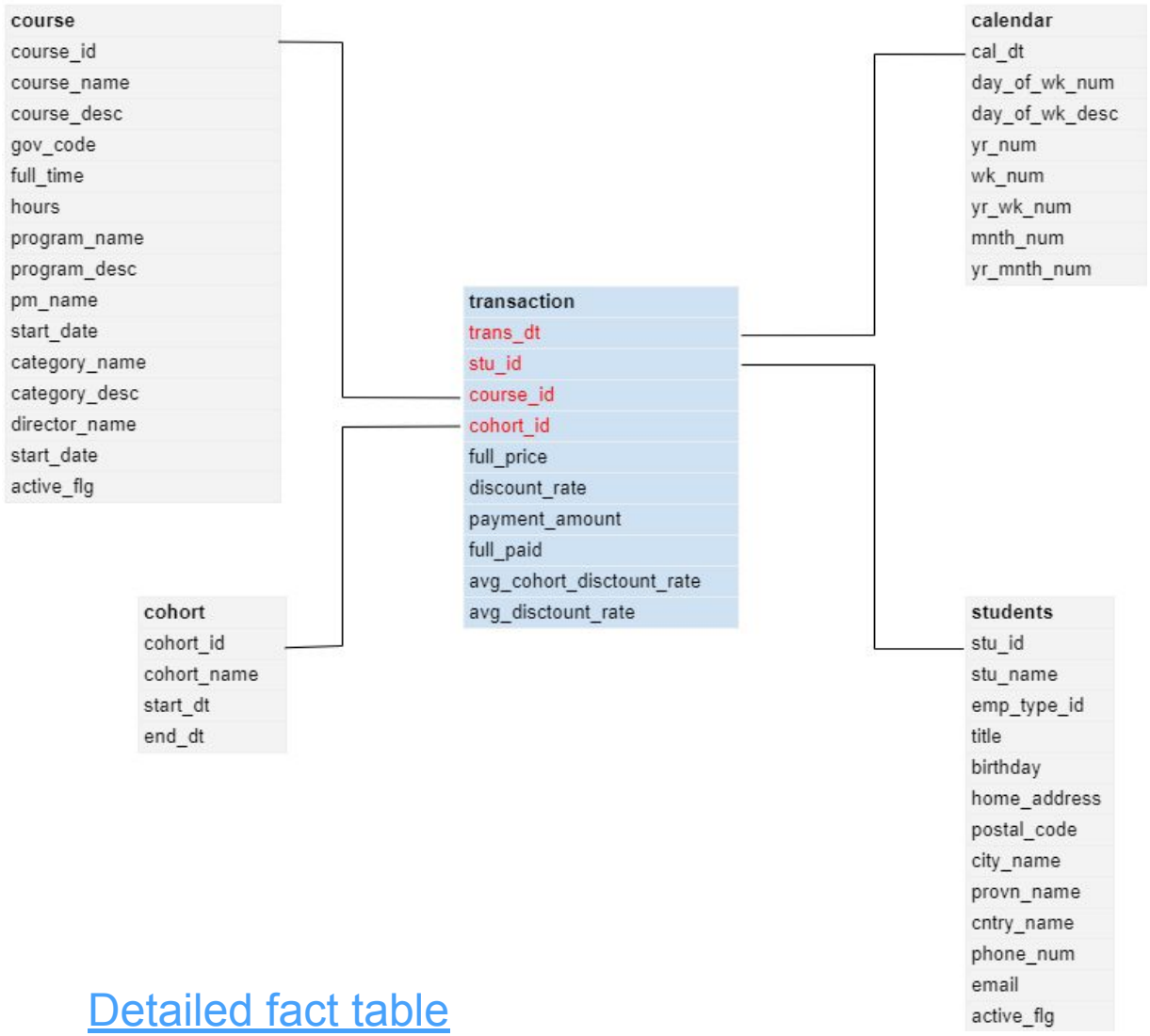
Step 3 Dimension Tables



1. Create new **course** dimension table by joining previous **course**,**program** and **category** tables.
   a. For the program manager and category director, we only use the name information from **employee** table, because we don't drill down in employee information in this project.
2. Keep **cohort** dimension as the same.
3. Create new **students** dimension table by join the **students** table with the **city** table to get full information for students.
4. Create a **calendar** dimension table including week, month and year information. The data for the calendar table are fixed, can be loaded at beginning of the project.

[Detailed dimension tables]

Step 4 Fact Table



[Detailed fact table]

1. The atomic row = trans_dt + stu_id + course_id + cohort_id
2. the ***full_price*** move from the previous transaction value.
3. The ***discount_rate*** is transferred from the discount type table.
4. the ***payment_amount*** is the sum of the payment（by date）of the atomic row from the previous transaction table.
5. A student may pay several times a day, and only the last transaction of payment is marked as full_paid, therefore the ***full_paid*** (True/False) column will only get the last value of the date from the previous transaction table. This column is for question 4.
6. ***avg_cohort_discount_rate*** is for question 5.
7. In order to make the end user query easier, we include the measures ***avg_discount_rate*** of the entire course in the fact table. The measure need to be calculated independently and joined to the transaction table afterward. We will discuss the transformation in the next lecture. This column is for question 6.

ETL

Prepared by WeCloudData

What is ETL

What is DDL

What is DML

Agenda.

ETL Concept

Design ETL

After data modelling, we need to build the pipeline how the system tables become the fact and dimension tables. This is the ETL process. The process converts the system raw tables and populate the converted data into the data model.

- ETL stand for Extract, Transform and Load. It represents the process of how the data is processed during the process.
- ETL and ELT:
  - ETL: Extract, Transform and Load.
  - ELT: Extract, Load, Transform.
  - It is not important to call a process ETL or ELT precisely. When we talk about the word ‘ETL’, it is not necessary to be ‘Extract, Transform and Load’, it also can be ‘Extract, Load, Transform’. We use ETL to represent the data pipeline process generally.

ETL Simple Mode

Design ETL

- deduplication and error fixing.
- transform to fix the tables in the data model.

OS data → **ETL** → Data Model

- This is the landing area for the original data from the operational system.
- keep original format, table names and column names.
- data validation run in this zone.
- mirror all the files.

- This is the analytic area to build data model.
- the data model includes fact and dimension tables.
- follow the naming convention.

Schema - SCHLND

Schema - SCHANL

Agenda.

What is ETL

What is DDL

What is DML

DDL

DDL (*Data Definition Language)*

It is used to define the structures like schema, database, tables, constraints etc. Examples of DDL are CREATE and ALTER statements. Simply put, Data model table creation is DDL.

DDL Build Entire Data Model and Infrastructures. Steps include:

- Create Schemas
- Create dimensions tables
- Create fact tables
- Create Control tables if needed

DDL Example

Example

```
CREATE OR REPLACE TABLE walsup.product_dim
(
    prod_key    Integer,
    prod_name   varchar(150),
    vol numeric(19,3),
    wgt numeric(19,3),
    brand_name  varchar,
    status_code varchar(30),
    status_code_name    varchar(30),
    category_key    integer,
    category_name   varchar(150),
    subcategory_key integer,
    subcategory_name    varchar(150),
    tlog_active_flg boolean,
    update_time timestamp default CURRENT_TIMESTAMP()
);
```

‘OR REPLACE’ will help to avoid creating error in case the table already exists.

Best Practice: add schema for each table as a habit. Tables’ naming conventions for table and col names.

Keys use integer

quantitative cols use default numeric(19,2) or (38,2).
Number in ‘( )’ presents:

- (Total number of digits, Number of digits after decimal)

code and text cols use varchar:
Number in ‘( )’ presents maximum number of characters to store. If no length specified, maximum allowed length (16,777,216).

flags use boolean, but sometimes use integer 1 or 0 to represent ‘true’ or ‘false’. This is based on the BI requirements.

‘update_time ‘is the col to record the table last updating time. Use timestamp as datatype, and ‘default *CURRENT_TIMPSTAMP()*’ function as a function to record the updating time automatically , no action needed in DML in this way.

DDL Description:

- Create DDL scripts for each zone.
- All the core tables creating queries are put in one script.
- Control table query in one script.
- DDL scripts are only used once when initializing the data model.

Try in system

Case Study (Continue)

1. Create data model Model in spreadsheet
2. Create DDL in Snowflake

What is ETL

What is DDL

What is DML

Agenda.

DML

DML(*Data Manipulation Language)*

It is used to manipulate data. Examples of DML are INSERT, UPDATE and DELETE statements. Simply put, Data model data transformation is DML.

DML Populate Data From One table to Another. Steps include:

- Populating onetime tables (*onetime tables are the tables only need populate once, like time table*)
- Populate dimension tables (*initial and incremental jobs*)
- Populate fact tables (*initial and incremental jobs*)
- In each Schema

DML Example - DIM

DDL and DML

```
USE SCHEMA walmart_dev.waletp;

TRUNCATE TABLE waletp.store_dim;
INSERT INTO waletp.store_dim
(
    store_key,
    store_name,
    status_code,
    status_cd_name
    )
SELECT
    store_key   AS store_key,
    store_desc  AS store_name,
    NULL AS status_code,
    NULL AS status_cd_name,
FROM wallnd.store
;
```

- It is a good habit to use schema name every time.
- But usually we will not hard code the schema name, because the database and schema name may be different in product system. Instead, we use **&{variable}** to input the variable from the environment. (review the knowledge Snowflake)
- If the table is designed as fully update, always truncate table each time before INSERT start.
- It is a good habit to use the target table columns' name as alias and the same columns' sequence in order to make sure you are updating to the right col.
- If there is no value to insert, use **NULL** to hold place.

Try in system

DML Example - FACT

DDL and DML

```
CREATE OR REPLACE TRANSIENT TABLE  wk_sales_inv_mk_period_stg1 as
SELECT
    b.per_group_id,
    b.per_group_desc,
    b.per_group_type,
    b.end_dt,
    yago_per_group_id,
    previous_per_group_id,
    store_lvl_2_key  AS  market_hier_key,
    store_lvl_2_desc AS market_hier_desc,
    a.prod_key,
    SUM(sales_amt) AS sales_amt,
    NULL AS avg_inventory_on_order_cost_amt,
    NULL AS avg_inventory_on_hand_total_cost_amt, --no dc, so dc_inventory_on_hand_cost_amt=0

FROM sales_inv_store a
    JOIN current_period_group_wk_ver b ON a.yr_wk_num = b.yr_wk_num AND b.per_group_id BETWEEN 1 AND 18
    JOIN calendar_period_group p ON b.per_group_id=p.per_group_id
    JOIN store_dim s ON a.store_key = s.store_key
GROUP BY 1,2,3,4,5,6,7,8,9
ORDER BY 1
;
```

- **Use transient table to staging results.**
- **To create a fact table may required to create several staging tables during the process.**

- **Be careful of using right type of JOIN.**

Try in system

Case Study (Continue)

1. Create script for one time dimension tables(calendar and time)
2. Create DML in Snowflake

Thank you

WeCloudData

500-80 Bloor Street West, Toronto
ON
www.weclouddata.com