Delta-Exam-Udemy-Questions

Question 1

Good job!

This answer most accurately depicts what happens to data when building a data warehouse.

Question 1:

Suppose that your organization decides to build a data warehouse for the first time. You have many different operational applications, but right now all of your reporting and analytics comes directly out of those applications. You decide that you need a data warehouse to support integrated reporting that will combine data from some or many of those source applications. After you successfully build and deploy your data warehouse, which of the following statements is most true about the data in your operational systems and your new data warehouse?

- **Your data will no longer be in the source systems and applications after being moved to the data warehouse.**
- **Your data will be copied from the source systems and applications to the data warehouse, but also retained in those original sources for operational purposes.**
- **Now, data will be written directly into the data warehouse by users as part of transactions such as hiring a new employee or a student registering for a course, rather than into the source systems and applications.**

Question 2

Good job!
BI is the discipline that makes use of data warehouse-produced data after the data is already brought into the data warehouse.

Question 2:

Congratulations! You have successfully built a data warehouse, and now it's time to put the data warehouse to use. Which related discipline is the one that will be used for your reporting and analytics?

- [x] **BI (business intelligence)**
- [ ] **ETL (extraction, transformation, and loading)**

Question 3

1. The university where you work has two different applications that contain student data: 1) a student tuition billing system and 2) a student academics and enrollment system.

   You are responsible for the integration work necessary to build a single "master list of students" in the data warehouse.

   A key part of your integration work will be to specify the data transformations necessary as part of the ETL.

   Relevant data fields in the **student tuition billing system**:

   Last Name: STRING (40)

   First Name: STRING (25)

   Academic Status: STRING (3) with the following permissible values:

   "FGS" – Full-time student in good standing

   "PGS" – Part-time student in good standing

   "FAP" – Full-time student on academic probation

   "PAP" – Part-time student on academic probation

   Relevant data fields in the **student academics and enrollment system**:

   Last Name: STRING (35)

   First Name: STRING (15)

   Enrollment Status: STRING (10) with the following permissible values:

   "Full-time"

   "Part-time"

   Academic Status: STRING (13) with the following permissible values:

   "Good standing"

   "Probation"

   YOUR ASSIGNMENT: Show the ETL transformations needed to unify the fields listed from the two different student-related applications into the data warehouse

Since we have two different systems with student data, we need to **extract** the data, **transform** it into a common format, and then **load** it into the data warehouse.

Step 1: Understand the Differences in Data

1. **Name Fields**
   - The "Last Name" and "First Name" fields exist in both systems but have different lengths.

- The data warehouse should store names with a standard length (let's choose the longest available).

2. **Academic Status Differences**
    - The **billing system** uses short codes (`FGS` , `PGS` , etc.).
    - The **enrollment system** uses full words (`Good standing` , `Probation`).
    - We need a unified format.
3. **Enrollment Status Differences**
    - The **billing system** combines enrollment and academic status into one field (`FGS` , `PGS` , etc.).
    - The **enrollment system** separates them (`Full-time` , `Part-time` and `Good standing` , `Probation`).
    - We need to split and map the combined values correctly.

Step 2: Define ETL Transformations

1. Extract Data

- Pull data from both sources.
- Read names, enrollment status, and academic status.

2. Transform Data

- **Name Fields Standardization**
    - Set `Last Name` to `STRING(40)` (since billing system allows 40).
    - Set `First Name` to `STRING(25)` (since billing system allows 25).
    - Trim excess characters if necessary.
- **Academic Status Mapping**
  Convert short codes from the **billing system** into full words to match the **enrollment system** format:

Billing System (Academic Status)	Enrollment System (Equivalent)
`FGS` (Full-time, Good standing)	`Full-time` + `Good standing`
`PGS` (Part-time, Good standing)	`Part-time` + `Good standing`
`FAP` (Full-time, Probation)	`Full-time` + `Probation`

Billing System (Academic Status)	Enrollment System (Equivalent)
`PAP` (Part-time, Probation)	`Part-time` + `Probation`

- **Final Data Warehouse Structure**
  The transformed data should have:

Last Name	First Name	Enrollment Status	Academic Status
`STRING(40)`	`STRING(25)`	`Full-time` / `Part-time`	`Good standing` / `Probation`

3. Load Data

- Insert the cleaned and transformed data into the **master list of students** in the data warehouse.

Final ETL Transformation Logic

Source	Transformation	Destination
Billing System: `Last Name (STRING 40)`	Keep as is	DW: `Last Name (STRING 40)`
Billing System: `First Name (STRING 25)`	Keep as is	DW: `First Name (STRING 25)`
Enrollment System: `Last Name (STRING 35)`	Pad/truncate to fit `STRING 40`	DW: `Last Name (STRING 40)`
Enrollment System: `First Name (STRING 15)`	Pad/truncate to fit `STRING 25`	DW: `First Name (STRING 25)`
Billing System: `Academic Status (FGS, PGS, FAP, PAP)`	Convert using mapping table	DW: `Enrollment Status` + `Academic Status`
Enrollment System: `Enrollment Status`	Keep as is	DW: `Enrollment Status`
Enrollment System: `Academic Status`	Keep as is	DW: `Academic Status`

Question 4

Good job!
Fact and dimension tables provide the foundation for the user access layer, and supply the dimensionality

Question 1:

Presuming your data warehouse will be used for reporting and online analytical processing (OLAP), your data in the user access layer should be:

- ( ) Structured and organized by the source systems that supply the data
- (•) **Structured dimensionally**

The correct answer is **"structured dimensionally"** because OLAP and reporting require data to be modeled in a way that supports fast queries, easy analysis, and intuitive reporting. Here's why:

1. Optimized for Analysis

- Dimensional modeling organizes data into **facts** (measurable business data) and **dimensions** (descriptive attributes).
- This makes queries faster and more intuitive compared to highly normalized OLTP structures.

2. Supports OLAP Operations

- Dimensional models facilitate **drill-down, roll-up, slice, and dice** operations in OLAP.
- Example: A sales fact table with dimensions like **time, product, region, and customer** allows easy analysis of sales trends.

3. Simplifies Queries

- Since dimensions are pre-joined with facts, complex joins (common in normalized schemas) are avoided.
- Querying a **star schema** (a common dimensional model) is simpler than querying an OLTP database.

4. Improves Performance

- Denormalized dimensional structures improve read performance by reducing joins and enabling indexing strategies.
- Aggregated data in fact tables allows for **quick reporting**.

5. Industry Standard for Data Warehouses

- **Ralph Kimball's methodology** promotes dimensional modeling as the best approach for OLAP-based data warehouses.
- Most BI tools are designed to work efficiently with **dimensional models**.

Thus, the **user access layer** in a data warehouse is structured **dimensionally** to optimize reporting and OLAP analysis.

Question 5

Good job!
This is correct

Question 2:

You are responsible for helping the business users in your organization understand the power of data warehousing, specifically with regards to dimensional analysis. Which key word usually signifies that you want to report or analyze only specific values within a dimension, rather than "slice up" the entire dimension?

- ○ By
- ◉ **For**
- ○ Select

The correct answer is **"For"** because it signifies that you want to report or analyze only **specific values within a dimension**, rather than slicing the entire dimension. Here's why:

1. "For" Implies Filtering a Specific Value

- When you say, **"Show me sales data for January 2025,"** you are filtering the **Time** dimension to a specific value (**January 2025**).
- You are not analyzing the entire time dimension but rather focusing on one particular period.

2. Difference Between "For" and "Slice"

- **Slicing** refers to selecting a **subset of a dimension** while keeping other dimensions intact (e.g., slicing the sales data for the "Electronics" category).
- **"For"** is more restrictive—it indicates filtering down to a **specific** value, not an entire category.
- Example: **"Sales for Electronics in Q1 2024"** vs. **"Slicing by Product Category"** (which might include multiple categories).

3. Common Usage in Business Questions

- Business users often phrase queries using **"for"** when they are interested in a particular value within a dimension.
- Examples:
  - **"Revenue for California stores"** (filters a specific region, not all regions).
  - **"Profit for Q3 2023"** (focuses on one quarter, not the whole year).
  - **"Customer churn rate for VIP members"** (targets a specific customer group).

Question 6

Good job!
This is correct because a GPA can't be meaningfully added, and thus is a non-additive fact

Question 3:

A Grade Point Average (GPA) is a good example of:

- ○ An additive fact
- ◉ **A non-additive fact**

Question 7

Good job!
Regardless of whether you are building a star schema or a snowflake schema, the home for dimensions will be your dimension tables

Question 4:

A business user comes to you and asks for an explanation of how the concepts of "facts" and "fact tables" relate to each other, as well as the similarities and differences between a star schema and snowflake schema. Then this user also asks about "dimensions" and "dimension tables." Which of the following is the most correct part of your answer as you explain dimensional analysis and data?

- "Fact" and "fact table" can be used interchangeably because they mean the same thing
- A star schema and a snowflake schema are structured exactly the same
- **You will find dimensions stored within dimension tables**

Question 8

Good job!
This is correct

Question 5:

One of your colleagues is helping you design your data warehouse, and asks what should be used for the primary keys of the new dimension tables. What is your answer to this question?

- Natural keys
- **Surrogate keys**

Question 9

Question 9.1

Design a transaction-grained fact table and the associated dimension tables to meet the following requirement:

1. The university has many different locations around campus that serve meals to students using their meal cards
2. We want to be able to analyze meal purchases by student, by date, and by location

Answer

A **transaction-grained fact table** means that each row in the fact table represents a **single meal purchase transaction**. This fact table will store measurable data (like how much was spent) and will be linked to **dimension tables** that describe different aspects of the transaction.

Step 1: Identify the Key Components

From the given requirements, we have three main things we want to analyze:
✅ **Student** – Who made the purchase?
✅ **Date** – When was the purchase made?
✅ **Location** – Where was the meal purchased?

Since we are recording transactions, we also need a **Fact Table** to store each meal purchase event.

Step 2: Designing the Tables
1. Fact Table: `Meal_Purchase_Fact`

This table stores each individual meal purchase transaction.

Meal_Purchase_ID	Student_ID	Date_ID	Location_ID	Amount_Spent	Meal_Type
1001	2001	20240310	301	5.00	Lunch
1002	2002	20240310	302	7.50	Dinner
1003	2003	20240309	301	6.00	Breakfast

- **Meal_Purchase_ID**: Unique ID for each transaction.
- **Student_ID**: Links to the Student Dimension.

- **Date_ID**: Links to the Date Dimension.
- **Location_ID**: Links to the Location Dimension.
- **Amount_Spent**: How much the student paid for the meal.
- **Meal_Type**: Breakfast, Lunch, or Dinner.

2. Dimension Table: `Student_Dim`

This table contains details about each student.

Student_ID	Name	Major	Year	Meal_Plan_Type
2001	Alex	Computer Science	2nd	Premium
2002	Ben	Mathematics	3rd	Standard
2003	Emma	Biology	1st	Basic

3. Dimension Table: `Date_Dim`

This stores information about dates for easy time-based analysis.

Date_ID	Full_Date	Day	Month	Year	Weekday
20240310	10-Mar-2024	10	March	2024	Sunday
20240309	09-Mar-2024	09	March	2024	Saturday

4. Dimension Table: `Location_Dim`

This contains information about meal service locations.

Location_ID	Location_Name	Building	Type
301	Campus Cafeteria	Main Hall	Dining Hall
302	Snack Corner	Library	Kiosk

Step 3: Why This Works?

1️⃣ The **fact table** captures every meal purchase transaction.
2️⃣ Each **dimension table** adds more details about the student, date, and location.
3️⃣ If we want to analyze **"Total meal purchases per location"**, we can group by

`Location_ID`.
4️⃣ If we want to see **“Student spending trends”**, we can filter by `Student_ID`.
5️⃣ The **Date Dimension** allows for easy **daily, monthly, and yearly analysis**.

Question 9.2

Design a periodic snapshot fact table and the associated dimension tables to meet the following requirement:

1. An important part of a faculty member’s position is publishing papers in academic journals
2. A faculty member will have, at any given point in time, some papers going through initial review; some papers going through final review; other papers that have been approved for publication but not published yet; and other papers that have been published
3. The university administration office conducts analysis at the end of each calendar month about faculty publications…specifically, how many papers each faculty member has at various stages at each “snapshot point in time”

Answer

A **periodic snapshot fact table** captures **summary data at regular time intervals** (e.g., end of each month). Unlike a transactional fact table, which records individual events, this table stores a **summary of faculty paper statuses at specific points in time**.

Step 1: Identify the Key Components

✅ **Faculty Member** – Who is publishing the papers?
✅ **Date (Snapshot Time)** – When is the data recorded? (End of each month)
✅ **Paper Status** – How many papers are at each stage (Initial Review, Final Review, Approved, Published)?

Step 2: Designing the Tables

1. Fact Table: `Faculty_Publication_Snapshot_Fact`

This table stores the count of papers for each faculty member at the end of each month.

Snapshot_ID	Faculty_ID	Date_ID	Papers_Initial_Review	Papers_Final_Review
1	101	20240229	2	1
2	102	20240229	3	2
3	101	20240331	1	2

📌 **Breakdown of Columns:**

- **Snapshot_ID**: Unique identifier for each snapshot entry.
- **Faculty_ID**: Links to the faculty member.
- **Date_ID**: Links to the date of the snapshot (end of month).
- **Papers_Initial_Review**: Count of papers in **Initial Review**.
- **Papers_Final_Review**: Count of papers in **Final Review**.
- **Papers_Approved**: Count of papers **approved but not published**.
- **Papers_Published**: Count of papers **officially published**.

2. Dimension Table: `Faculty_Dim`

This table stores details about faculty members.

Faculty_ID	Name	Department	Rank	Research_Area
101	Dr. Smith	Computer Science	Professor	Artificial Intelligence
102	Dr. Brown	Physics	Associate Professor	Quantum Mechanics

3. Dimension Table: `Date_Dim`

This stores information about the snapshot dates (end of each month).

Date_ID	Full_Date	Month	Year	Quarter
20240229	29-Feb-2024	Feb	2024	Q1
20240331	31-Mar-2024	Mar	2024	Q1

Step 3: Why This Works?

✅ **Tracks Changes Over Time**: The **periodic snapshot** captures how the number of papers in each status changes month-to-month.
✅ **Enables Trend Analysis**: University admins can analyze **publication trends** over time, such as:

Question 10

Question 10.1

Design a factless fact table and the associated dimension tables to meet the following requirement:

1. The university annually holds a three-day orientation program for incoming freshmen
2. Seniors are encouraged to participate in the orientation program, along with the university administration personnel
3. Each senior can participate in part or all of the orientation program…e.g., all three days, or just the 1st day, or just the 2nd and 3rd days, etc.
4. To help plan activities, each senior signs up in an online survey system for which day(s) he/she will be participating
5. The online survey system sends this information into the university data warehouse, where it is stored for analysis in a factless fact table

Answer

- The university holds a **3-day orientation program**.
- Seniors can **choose which days to participate**.
- The university wants to **analyze sign-ups** to help plan activities.
- Since there are **no measurable facts like amounts or counts**, we use a **factless fact table** to record participation.

  Step 1: Identifying Dimensions

The main elements of this scenario are:
✅ **Senior (Student)** – Who is signing up?

✅ **Date (Orientation Day)** – Which day are they attending?
✅ **Event (Orientation Program)** – The event itself.

Step 2: Designing the Tables

1. Factless Fact Table: `Senior_Orientation_Fact`

This table captures the participation of each senior for each orientation day.

Student_ID	Date_ID	Event_ID
1001	20250901	1
1001	20250902	1
1002	20250902	1
1003	20250903	1

2. Dimension Table: `Student_Dim`

This table stores information about seniors participating.

Student_ID	Name	Major	Year
1001	Alex	Biology	Senior
1002	Ben	Math	Senior
1003	Chris	CS	Senior

3. Dimension Table: `Date_Dim`

This stores the **dates of the orientation program**.

Date_ID	Full_Date	Day	Month	Year
20250901	01-Sep-2025	1	Sep	2025
20250902	02-Sep-2025	2	Sep	2025
20250903	03-Sep-2025	3	Sep	2025

4. Dimension Table: `Event_Dim`

This represents the **orientation program**.

Event_ID	Event_Name	Duration	Organizer
1	Orientation Program	3 Days	Admin Team

Question 10.2

Design an accumulating snapshot fact table and the associated dimension tables to meet the following requirement:

1. An important part of a faculty member's position is pursuing grants from various sources
2. For purposes of this exercise, assume that every grant goes through exactly the same process, regardless of source
3. A faculty member applies for a grant, for a specific amount of money
4. Each grant goes through an initial approval step; if approved, the amount of money might be reduced from the original request
5. Each grant then goes through a final approval step; if approved, the amount of money might be reduced from the amount that entered the final approval step
6. The grant money for an approved grant is eventually paid to the faculty member

Answer

An **accumulating snapshot fact table** is used to track **the progress of a process that moves through well-defined stages**.

Unlike a **transactional fact table** (which captures individual events) or a **periodic snapshot fact table** (which records periodic summaries), an accumulating snapshot fact table **gets updated** as the process advances.

Understanding the Scenario

- Faculty members apply for **grants**.
- Each grant **moves through multiple stages** (Application → Initial Approval → Final Approval → Payment).
- The **funding amount may change** at each step.
- The process has **clear milestones**, making it ideal for an **accumulating snapshot fact table**.

Step 1: Identify the Key Components

✅ **Grant** – The main entity moving through the process.
✅ **Faculty Member** – Who applied for the grant?
✅ **Dates** – When did the grant reach each stage?
✅ **Grant Amounts** – How much money was requested, approved, and paid?

Step 2: Designing the Tables

1. Accumulating Snapshot Fact Table: `Grant_Approval_Fact`

Each row represents **one grant**, and we update the row as the grant moves through the process.

Grant_ID	Faculty_ID	Applied_Date_ID	Initial_Approval_Date_ID	Final_Approval_D
1001	201	20240301	20240310	20240320
1002	202	20240305	20240315	NULL

📌 Breakdown of Columns:

- **Grant_ID**: Unique identifier for each grant.
- **Faculty_ID**: Links to the faculty member applying.
- **Applied_Date_ID**: Date when the grant was applied for.
- **Initial_Approval_Date_ID**: Date of the first approval (NULL if still pending).
- **Final_Approval_Date_ID**: Date of the final approval (NULL if not yet approved).
- **Payment_Date_ID**: Date when funds are disbursed (NULL if not yet paid).
- **Requested_Amount**: Original grant request amount.
- **Initial_Approved_Amount**: Amount approved in the first review (NULL if still in progress).
- **Final_Approved_Amount**: Amount approved after final review (NULL if still in progress).
- **Paid_Amount**: Final disbursed amount (NULL if not yet paid).

✅ Why is this an accumulating snapshot?

- The row **stays the same** but gets **updated** as the grant progresses.
- At any time, we can see **where each grant is in the process**.

2. Dimension Table: `Faculty_Dim`

This table stores details about faculty members.

Faculty_ID	Name	Department	Rank
201	Dr. Smith	Computer Science	Professor
202	Dr. Brown	Physics	Associate Professor

3. Dimension Table: `Date_Dim`

This stores date-related information for tracking approvals and payments.

Date_ID	Full_Date	Day	Month	Year
20240301	01-Mar-2024	01	March	2024
20240310	10-Mar-2024	10	March	2024
20240320	20-Mar-2024	20	March	2024

Question 11

Question 11.1

Mary Johnson has been enrolled in the university's PhD program while teaching, and has now earned her PhD
Mary is now being considered for promotion next academic year but continues to teach as a Lecturer

What does the relevant data in FACULTY_DIM for Mary Johnson now look like?

Mary Johnson is approved for promotion to Assistant Professor because she now has a PhD.
What does the relevant data in FACULTY_DIM for Mary Johnson now look like?

Answer

The **FACULTY_DIM** table is a **slowly changing dimension (SCD)** because faculty members' details (such as rank and degree) change over time. The way we handle these changes depends on the **SCD type** used in the data warehouse.

For these scenarios, we assume **SCD Type 2**, which keeps historical records by inserting a new row for each significant change while maintaining previous records.

Scenario 1: Mary Johnson earns her PhD but remains a Lecturer

Faculty_ID	Name	Rank	Degree	Status	Start_Date	End_Date	Is_Current
1001	Mary Johnson	Lecturer	MSc	Active	2020-08-01	2025-06-30	No
1002	Mary Johnson	Lecturer	PhD	Active	2025-07-01	NULL	Yes

📌 **What changed?**

- A **new row is added** because Mary **earned her PhD**.
- The **previous record** is closed (`End_Date = 2025-06-30`).
- The **new row** has an updated **Degree = PhD** and remains **Lecturer**.

Scenario 2: Mary is promoted to Assistant Professor

Faculty_ID	Name	Rank	Degree	Status	Start_Date	End_Date	Is_Current
1001	Mary Johnson	Lecturer	MSc	Active	2020-08-01	2025-06-30	No
1002	Mary Johnson	Lecturer	PhD	Active	2025-07-01	2026-06-30	No
1003	Mary Johnson	Assistant Professor	PhD	Active	2026-07-01	NULL	Yes

📌 **What changed?**

- A **new row is inserted** for the promotion to **Assistant Professor**.
- The **previous Lecturer (PhD) record is closed** (`End_Date = 2026-06-30`).
- The **new row** reflects **Rank = Assistant Professor**.

Question 11.2

Mary Johnson has been enrolled in the university's PhD program while teaching and (unlike Question 1) is still working on her PhD
Mary is promoted from Lecturer to Senior Lecturer (again, while still working on her PhD)

What does the relevant data in FACULTY_DIM for Mary Johnson now look like?

Mary Johnson now completes her doctoral studies and earns her PhD…but is still teaching as a Senior Lecturer

What does the relevant data in FACULTY_DIM for Mary Johnson now look like?

Answer

Scenario 1: Mary is still pursuing her PhD but gets promoted to Senior Lecturer

Faculty_ID	Name	Rank	Degree	Status	Start_Date	End_Date	Is_Current
1001	Mary Johnson	Lecturer	MSc	Active	2020-08-01	2025-06-30	No
1002	Mary Johnson	Senior Lecturer	MSc	Active	2025-07-01	NULL	Yes

📌 **What changed?**

- A **new row is inserted** for the promotion to **Senior Lecturer**.
- The **degree remains MSc** because she is still working on her PhD.

Scenario 2: Mary completes her PhD but remains a Senior Lecturer

Faculty_ID	Name	Rank	Degree	Status	Start_Date	End_Date	Is_Current
1001	Mary Johnson	Lecturer	MSc	Active	2020-08-01	2025-06-30	No
1002	Mary Johnson	Senior Lecturer	MSc	Active	2025-07-01	2026-06-30	No
1003	Mary Johnson	Senior Lecturer	PhD	Active	2026-07-01	NULL	Yes

📌 **What changed?**

- A **new row is inserted** for earning a **PhD**.
- The **rank remains Senior Lecturer**, but now **Degree = PhD**.
- The previous record is **closed with an End_Date = 2026-06-30**.

Question 12

Good job!
Correct - dimension table surrogate keys are needed for fact tables, and therefore must be processed first.

Question 1:

When designing your ETL, which types of tables do you need to process first as part of an incremental ETL run"?

- ○ Fact tables
- ◉ **Dimension tables**
- ○ Fact tables and dimension tables can be processed at the same time; it doesn't matter which is processed first.

Question 13

Good job!
This is correct, based on what was explained in the lecture videos

Question 2:

One of your colleagues is helping you with ETL design, and is confused about the proper order of the steps to follow. You walk to the whiteboard and write down the order of steps. Which of the following answers is what you will write?

- ○ 1) Data transformation; 2) data preparation; 3) process and load Type 1 and 2 changes; 4) process and load new dimension table data; 5) process and load new fact table data
- ○ 1) Data preparation; 2) process and load new dimension table data; 3) process and load Type 1 and 2 changes; 4) process and load new fact table data; 5) data transformation
- ◉ **1) Data preparation; 2) data transformation; 3) process and load new dimension table data; 4) process and load Type 1 and 2 changes; 5) process and load new fact table data**

Question 14

Good job!
This is correct, because now we will have "2 versions of the same dimensional data" based on the rules of the Type 2 change

Question 3:

When doing ETL processing for a Type 2 SCD (slowly changing dimension) change, you will:

- ○ Overwrite and replace an existing value in a dimension table row
- ○ Add a new dimension table row with the new data and a new surrogate key, and then change the surrogate key/primary key of the old dimension table row to be the same as the new row's surrogate key
- ◉ **Add a new dimension table row with the new data and a new surrogate key, and leave the old/existing dimension table row exactly as is...except for possibly changing the "Current" flag to "N" and/or updating the expiration date for that row**

Question 15

Good job!
This is correct, as explained in the lecture video. You still need to carefully design which tables are able to be processed in parallel based on relative sizes and other factors

Question 4:

Which of the following statements is most true for when you are designing each incremental ETL job for your organization's new data warehouse?

- ○ You can only process one dimension table at a time, but you can process multiple fact tables in parallel with each other
- ○ You can only process one fact table at a time, but you can process multiple dimension tables in parallel with each other
- ◉ **You can process dimension tables in parallel with each other, and after all of your dimension tables have been processed you can then process fact tables in parallel with each other**

Question 16

Good job!
This is true based on the lecture video

Question 1:

One of your company executives asks you "what's the deal with cloud computing and data warehousing?" Which of the following statements should be part of your answer?

- ○ Today, data lakes are typically built on cloud platforms, while data warehouses are almost always built in corporate data centers.
- ◉ **Today, data warehouses are found in both corporate data centers and in the cloud, but the numbers of DWs being built in the cloud is steadily increasing**
- ○ Today, cloud-based data warehouses are commonly being migrated from cloud platforms to corporate data centers

Question 17

Good job!
This is true based on the lecture video

Question 2:

When building a data warehouse (or data lake) in a cloud platform:

- ○ Data "egress" (accessing data) is the lowest-cost component to your overall pricing and cost of ownership
- ◉ **You need to carefully plan for and watch the costs associated with your data "egress" (accessing data)**
- ○ You don't need to worry about data "ingress" or "egress" because the cloud hosting companies provide those services for free

Question 18

Good job!

This is true, based on the lecture video

Question 3:

Suppose that one of your colleagues asks you to summarize the relationship between how many of your users will typically view the data from your data warehouse versus the underlying platform. Which of the following statements is most correct for your answer?

- ○ The user-facing view of a data center-hosted data warehouse is typically dimensional, but a cloud-hosted data warehouse is typically not dimensional.
- ◉ **Regardless of whether a data warehouse is built "in the cloud" or a corporate data center, users will typically still see the data dimensionally.**

Question 19

Good job!

The two categories of data processing are analytical and operational processing.

Question 1:

What are usually the two main purposes of processing data in a company?

- ○ Transactional and operational
- ◉ **Analytical and operational**
- ○ Analytical and transactional

Question 20

Good job!

This is analytical data processing, since data is analyzed, aggregated and used for analytical purposes.

Question 2:

What is *not* an example of operational data processing?

- ○ A banking app receives and processes a transfer request from a user
- ○ An employee in a warehouse scans a product that was order and is picked
- ◉ **A regional manager calculates how many items an employee has picked in average per hour**
- ○ A billing system calculates the monthly charges for a customer and sends the invoice per email

Question 21

Good job!

Yes, OLTP stands for Online Transactional Processing

Question 3:

What is a term used to describe *operational data processing*?

- ○ OLAP
- ◉ **OLTP**

Question 22

Good job!

Question 4:

What is *not* a key characteristic of a data warehouse?

- Centralized location for data
- Fast query performance
- User-friendly
- Data recovery system

Question 23

Good job!
A data lake would be a good solution for the large volume of unstructured data.

Question 1:

You are working in a logistics company that uses different IoT devices in unstructured formats. What would you rather choose as a centralized location for the hundreds of millions of datasets?

- Data Lake
- Data Warehouse

Question 24

Good job!

A datawarehouse is a good fit for that. This is because a report usually needs to have structured data from tables and ideally they have a fast query performance to visualize the data quickly and a high user-friendliness. Therefore a data warehouse is the better fit.

Question 2:

You want to create a report for the finance department. The data comes from different sources. What would you rather create and establish as a centralized data source for this report?

- ○ Data lake
- ◉ **Data warehouse**

Question 25

Good job!

This was discussed in Lecture 14: **Staging area** ›

Question 1:

What is not a reason for a staging layer?

- ◉ **Increasing the performance of the access layer**
- ○ Spending short time on productive systems
- ○ Getting data into tables in a relational database

Question 26

Good job!
Exactly, this is where we apply the transformation logic and load the transformed data.

Question 2:

Where do we apply most of the transformations including data cleaning in the ETL process?

- ○ From the data sources to the staging layer
- ◉ **From the staging layer to the core layer**
- ○ From the core layer to the data marts

Question 27

Good job!
Exactly, this is usually not a good reason to built a dedicated data warehouse. We usually want to analyze data over time and different periods usually don't constitute a different use-case.

This was discussed in Lecture 16: **Data Marts** ›

Question 3:

What is not a good reason to create a dedicated data mart?

- ○ Different user groups
- ○ Different tools
- ◉ **Different periods**
- ○ Different departments

Question 28

Good job!

Question 1:

What is the main reason for improved query performance in Cubes?

- Less latency due to multi-dimensional approach
- Better query optimizer
- Precalculated (aggregated) values

Question 29

Good job!

Question 2:

What is the key idea for improving performance with in-memory databases?

- Defining multiple dimension and precalculating the relevant values.
- Eliminating response time from disc by processing all data directly in memory.
- Optimizing queries with better query optimizers

Question 30

Good job!

Question 1:

What can reduce the query performance in a snowflake schema?

- (selected) **Many tables and with that more joins**
- Less data redundancy
- More data redundancy
- Few tables with less joins

Question 31

Good job!

This was discussed in Lecture 27: **Snowflake schema** ›

Question 2:

What is usually not a reason to create a snowflake schema?

- Less data storage
- (selected) **More usability**
- Can be easier to maintain and update data in some cases

Question 32

Good job!

This was discussed in Lecture 27: **Snowflake schema** >

Question 3:

Which of the two schema types is more normalized?

- (•) **Snowflake schema**
- ( ) Star schema
- ( ) There is no difference in terms of normalization

Question 33

Good job!

Question 4:

What is not a goal when creating a star schema?

- ( ) Improving performance with read queries
- (•) **Improving performance write operations**
- ( ) Improving usablity

Question 34

Good job!

Yes we can aggregate the items to see how many items are in stock in total. But we cannot aggregate them across the date dimension.

Question 1:

This table displays the number of items that are in stock in the company's warehouse for each product in each day.

Date	Product_FK	Items in stock
20-07-2022	1	15
20-07-2022	2	1
20-07-2022	3	18
20-07-2022	4	6
21-07-2022	1	34
21-07-2022	2	14
21-07-2022	3	34
21-07-2022	4	10
22-07-2022	1	27
22-07-2022	2	14
22-07-2022	3	31
22-07-2022	4	12
23-07-2022	1	13
23-07-2022	2	4
23-07-2022	3	15
23-07-2022	4	8

Is the fact "Items in stock" additive, semi-additive or non-additive?

- Non-additive
- **Semi-additive**
- Additive

Question 35

Good job!
We can add up the products sold across all dimensions and we get a number that makes sense - total number of items sold.

Question 2:

This table displays the number of items that have been sold by the company for each product in each day.

Date	Product_FK	Quantity sold
20-07-2022	1	15
20-07-2022	2	24
20-07-2022	3	0
20-07-2022	4	19
21-07-2022	1	8
21-07-2022	2	21
21-07-2022	3	22
21-07-2022	4	17
22-07-2022	1	6
22-07-2022	2	32
22-07-2022	3	24
22-07-2022	4	22
23-07-2022	1	34
23-07-2022	2	21
23-07-2022	3	0
23-07-2022	4	33

Is the fact "Quantity sold" additive, semi-additive or non-additive?

- ○ Non-additve
- ○ Semi-additive
- ◉ **Additive**

Question 36

Good job!

Adding up any discounts across any dimension (either different products or different days) does not result in a meaningful number.

Question 3:

This table displays the discount of each products in a given day that the company offers.

Date	Product_FK	Discount
20-07-2022	1	0%
20-07-2022	2	21%
20-07-2022	3	9%
20-07-2022	4	23%
21-07-2022	1	15%
21-07-2022	2	19%
21-07-2022	3	14%
21-07-2022	4	13%
22-07-2022	1	17%
22-07-2022	2	19%
22-07-2022	3	3%
22-07-2022	4	7%
23-07-2022	1	9%
23-07-2022	2	7%
23-07-2022	3	9%
23-07-2022	4	16%

Is the fact "Discount" additive, semi-additive or non-additive?

- (•) **Non-additve**
- ( ) Semi-additive
- ( ) Additive

Question 37

Good job!
The data is aggregated periodically (each day = standard period).

Question 1:

This table displays the number of items that are in stock in the company's warehouse for each product in each day.

Date	Product_FK	Items in stock
20-07-2022	1	15
20-07-2022	2	1
20-07-2022	3	18
20-07-2022	4	6
21-07-2022	1	34
21-07-2022	2	14
21-07-2022	3	34
21-07-2022	4	10
22-07-2022	1	27
22-07-2022	2	14
22-07-2022	3	31
22-07-2022	4	12
23-07-2022	1	13
23-07-2022	2	4
23-07-2022	3	15
23-07-2022	4	8

What type of fact table is it?

- Transactional fact table
- **Periodic Snapshot fact table**
- Accumulating snapshot table

Question 38

Good job!

Yes, one row represents one sale (=transaction).

Question 2:

Below you see the star schema of a data model with dimensions and facts.

Dim_Date
Date_ID
Year
Quarter
Month
Day

Fact_Sales
SalesID
Date_ID
Quantity
SalesAmount
Item_ID
Location_ID
Brand_name
Category_name
Location_id

Dim_Location
Location_ID
Country
State
City
Location_manager

Dim_Item
Item_ID
Item_Name
Brand_name
Category_name

As what type of fact table would you identify the Fact_Sales table?

- (●) **Transactional**
- ( ) Periodic Snapshot
- ( ) Accumulative Snapshot

Question 39

Good job!

Question 1:

Why do we even use dimension in our data model?

- [ ] It helps with update and insert operations
- [x] **It helps with performance and usability**
- [ ] It helps to deal with heterogeneous data

Question 40

Good job!

This was discussed in Lecture 46: **Date dimensions** ›

Question 2:

What is a common key format for date dimensions?

- [ ] We commonly use a standard (and meaningless) surrogate key like 1, 2, 3, 4,...
- [x] **We commonly use a meaningful surrogate key in the format YYYYMMDD.**
- [ ] We commonly use no surrogate key but just the date column in the format "YYYY-MM-DD".

Question 41

Good job!

Yes. We create multiple timestamp keys that can be all used together with our date dimension. That means our date dimension can play different roles - depeinding on which key is used to create the relationshipt to the date table.

Question 3:

In your order processing fact table you have multiple timestamps for different steps of the order processing. The analysts/business users want to analyze the data based on all these different timestamps. How can you fulfill their requirement?

- (selected) **Using a role-playing dimension**
- Using a degenerate dimension
- Using a junk dimension
- Using a conformed dimension

Question 42

Good job!

A perfect partition of the history we can achieve using SCD Type 2 by adding a new row for every update/insert of a new dimension value.

This was discussed in Lecture 57: **Type 2 - Additional row** ›

Question 1:

What type of slowly changing dimensions should we use if the changes can be more frequent and occur not really predictable?

- ( ) Type 0
- ( ) Type 1
- (•) **Type 2**
- ( ) Type 3

Question 43

Good job!

This was discussed in Lecture 58: **Administrating Type 2 dimensions** ›

Question 2:

If there is are columns like "Effective date" and "Expiry date" in a given dimension. What type of slowly changing dimension are we dealing with?

- ( ) Type 0
- ( ) Type 1
- (•) **Type 2**
- ( ) Type 3

Question 44

Good job!

Question 3:

Given that it is not important to maintain any history of changes of a given dimension but we prefer to use a simple method, what type of SCD can you implement?

- (x) **Type 1**
- ( ) Type 2
- ( ) Type 3