

# L11 - Database Optimization

**Due** Apr 29 at 11:59pm

**Points** 40

**Questions** 14

**Available** after Apr 28 at 9am

**Time Limit** None

**Allowed Attempts** Unlimited

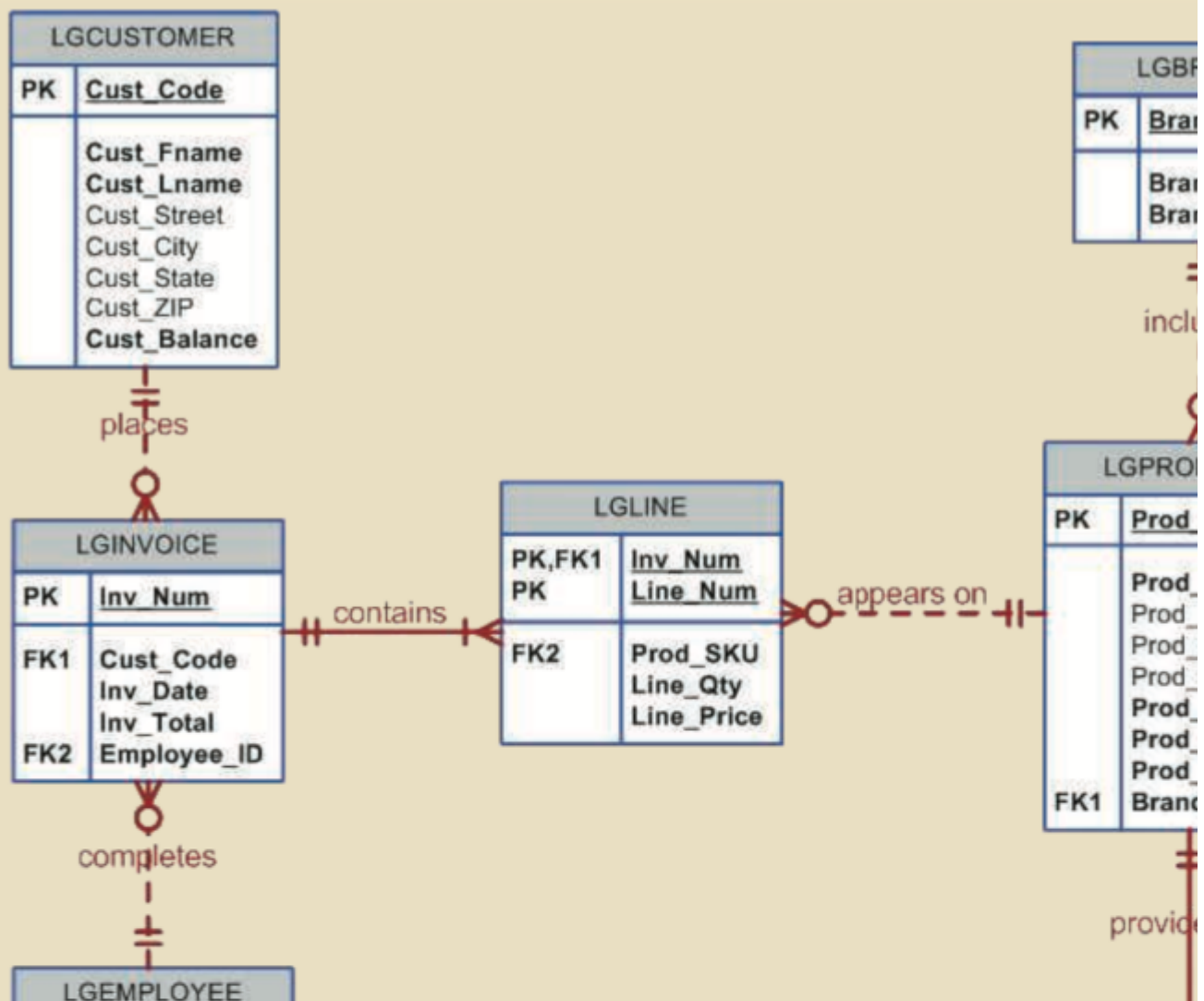
## Instructions

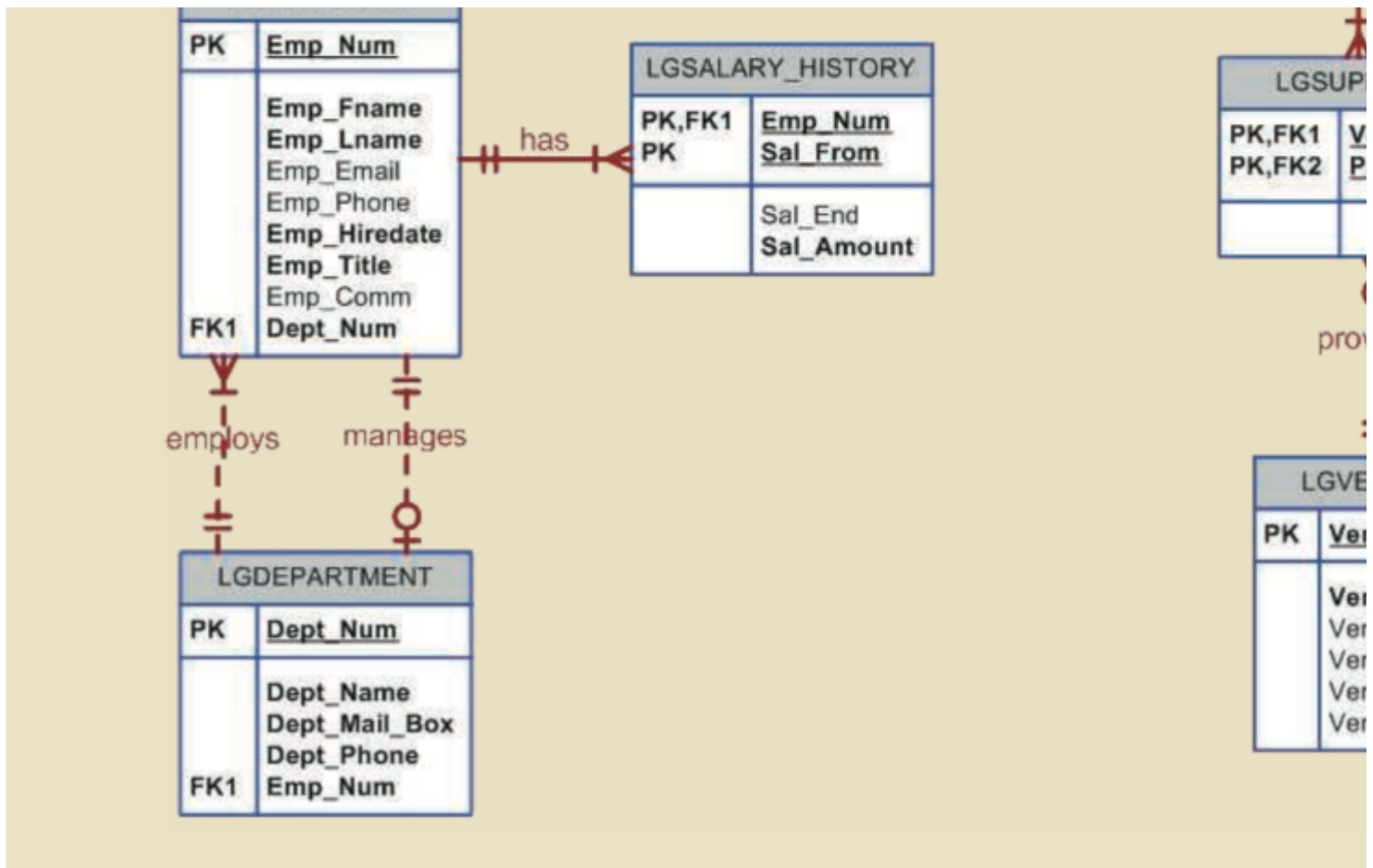
In this lab we will be looking at some aspects of query optimization and using that knowledge to analyze the Largeco test database.

The SQL files for this lab are split in two:

1. Largeco DDL commands: COMP\_420\_Spring\_2021\_Lab\_11\_largeco\_ddl.sql
2. Largeco data inserts: COMP\_420\_Spring\_2021\_Lab\_11\_largeco\_inserts.sql

FIGURE P8.19 THE LARGECO ERD





Take the Quiz Again

## Attempt History

	Attempt	Time	Score
LATEST	<a href="#">Attempt 1</a>	1,922 minutes	34 out of 40

⚠ Correct answers are hidden.

Score for this attempt: **34** out of 40

Submitted Apr 29 at 5:37pm

This attempt took 1,922 minutes.

## Part 1: Insert and Analyze

In this section you will populate the Largeco database, optimize it through the use of proper indexing, and perform some basic analysis.

## Question 1

10 / 10 pts

Referencing the Largeco ERD above, add ALL the appropriate primary and foreign key statements using a series of ALTER TABLE commands (10pts).

**Note: pay attention to compound keys.**

**Please paste your alter table DDL code in the box below.**

Your Answer:

use largeco;

ALTER TABLE LGCUSTOMER add primary key (Cust\_Code);

ALTER TABLE LGINVOICE add primary key (Inv\_Num);

ALTER TABLE LGINVOICE add constraint fkref foreign key (Cust\_Code) references LGCUSTOMER (Cust\_Code);

ALTER TABLE LGINVOICE add constraint fkref2 foreign key (Employee\_ID) references LGEMPLOYEE (Emp\_Num);

ALTER TABLE LGLINE add primary key (Inv\_Num);

ALTER TABLE LGLINE add constraint fk foreign key (Inv\_Num) references LGINVOICE (Inv\_Num);

ALTER TABLE LGLINE add primary key (Line\_Num);

ALTER TABLE LGLINE add constraint fk4 foreign key (Prod\_SKU) references LGPRODUCT (Prod\_SKU);

ALTER TABLE LGBRAND add primary key (Brand\_ID);

ALTER TABLE LGPRODUCT add primary key (Prod\_Sku);

ALTER TABLE LGLINE add constraint fk5 foreign key (Brand\_ID) references LGBRAND (Brand\_ID);

```
ALTER TABLE LGEMPLOYEE add primary key (Emp_Num);

ALTER TABLE LGEMPLOYEE add constraint fk6 foreign key (Dept_Num)
references LGDEPARTMENT (Dept_Num);

ALTER TABLE LGSALARY_HISTORY add primary key (Emp_Num);

ALTER TABLE LGSALARY_HISTORY add constraint fk7 foreign key
(Emp_Num) references LGEMPLOYEE (Emp_Num);

ALTER TABLE LGSALARY_HISTORY add primary key (Sal_From);

ALTER TABLE LGDEPARTMENT add primary key (Dept_Num);

ALTER TABLE LGDEPARTMENT add constraint fk8 foreign key
(Emp_Num) references LGEMPLOYEE (Emp_Num);

ALTER TABLE LGSUPPLIES add primary key (Vend_ID);

ALTER TABLE LGSUPPLIES add constraint fk2 foreign key (Vend_ID)
references LGVENDOR (Vend_ID);

ALTER TABLE LGSUPPLIES add primary key (Prod_SKU);

ALTER TABLE LGSUPPLIES add constraint fk3 foreign key (Prod_Sku)
references LGPRODUCT (Prod_Sku);

ALTER TABLE LGVENDOR add primary key (Vend_ID);
```

## Question 2

6 / 6 pts

Reverse engineer your database in MySQL Workbench and **upload an image of it that clearly shows your primary key indexes**. You may need to click the little triangle at the bottom of each table to make them visible.

 [11#2.png \(https://cilearn.csuci.edu/files/2852380/download\)](https://cilearn.csuci.edu/files/2852380/download)

## Question 3

5 / 6 pts

As the database administrator for Largeco, you frequently need to run the following query:

```
SELECT * FROM lgproduct  
WHERE p_descript = {description};
```

“description” is a generic product description in this case.

Answer the following questions:

1. Would an index help this query? Why or why not?
2. What field should be indexed?
3. Write a query to create an index on your selected field.

Your Answer:

1. No, because it's description won't be unique.
2. Product\_SKU because it is uniquely identified.
3. CREATE INDEX indexprod\_sku ON lgproduct (prod\_sku);

uniqueness and indexing are not really related in this way, you can certainly index fields that are not unique

## Part 2: Basic Statistics

Before proceeding with this section, please run the Largeco insert script SQL file.

### Question 4

2 / 2 pts

Run an ANALYZE command to update the database statistics.

***Please upload a screenshot of your query and its result.***

 [11#4.JPG \(https://cilearn.csuci.edu/files/2852812/download\)](https://cilearn.csuci.edu/files/2852812/download)

### Question 5

2 / 2 pts

Write a query using MySQL's INFORMATION\_SCHEMA that returns the total number of rows in the entire Largeco database.

***Please upload a screenshot of your query and its result with your name and the date embedded.***

 [11#5.JPG \(https://cilearn.csuci.edu/files/2852813/download\)](https://cilearn.csuci.edu/files/2852813/download)

### Question 6

1 / 2 pts

Write a query using MySQL's INFORMATION\_SCHEMA that returns the size, in 16KB pages (1024 bytes in KB), of all the tables in Largeco.

***Please upload a screenshot of your query and its result with your name and the date embedded.***

 [11#6.JPG \(https://cilearn.csuci.edu/files/2852814/download\)](https://cilearn.csuci.edu/files/2852814/download)

### Question 7

2 / 2 pts

Write a query using MySQL's INFORMATION\_SCHEMA that returns the table with the largest index (as determined by size in memory).

***Please upload a screenshot of your query and its result with your name and the date embedded.***

 [11#7.JPG \(https://cilearn.csuci.edu/files/2852815/download\)](https://cilearn.csuci.edu/files/2852815/download)

## Part 3: Query Analysis

Using the EXPLAIN command, answer the following questions.

### Question 8

1 / 1 pts

Remove the cust\_code foreign key constraint AND index from the LGINVOICE table.

For this question and the next, analyze the query:

```
SELECT * FROM lginvoice;
```

How many rows does the query return?

Incorrect

### Question 9

0 / 2 pts

Based on the previous question, what type of access method type did the query use and what type of access strategy (Table Scan, Table Access, or Index Scan) did it use?

Access method:

Access strategy:

**Answer 1:**

row ID

**Answer 2:**

Index Scan

Add the cust\_code foreign key constraint back onto the LGINVOICE table.

For the next three questions, analyze the query:

```
SELECT * FROM lginvoice JOIN lgcustomer USING(cust_code)
```

**Incorrect****Question 10****0 / 1 pts**

For the query to be analyzed, how many rows were examined from LGINVOICE?

3,351

**Incorrect****Question 11****0 / 1 pts**

For the query to be analyzed, what type of access method was used for LGCUSTOMER?

table scan



**Question 12****2 / 2 pts**

What effect did adding the foreign key index on cust\_code have on the query and why?

Your Answer:

Foreign key constraint improve performance at the time of reading data but at the same time it slows down the performance at the time of inserting / modifying / deleting data.

In case of reading the query, the optimizer can use foreign key constraints to create more efficient query plans as foreign key constraints are pre declared rules. This usually involves skipping some part of the query plan because for example the optimizer can see that because of a foreign key constraint, it is unnecessary to execute that particular part of the plan.

**Question 13****1 / 1 pts**

Analyze the query:

```
SELECT * FROM lgcustomer WHERE 1 = 0;
```

How did MySQL react to this query?

Your Answer:

All the field values were returned as nulls as the query prevented the output.

**Question 14****1 / 2 pts**

Analyze the query:

```
SELECT * FROM lgcustomer WHERE 1 = 1;
```

1. Did MySQL execute the WHERE clause?
2. How do you know?

Your Answer:

1. The WHERE clause was executed as a means to return all "true" rows, meaning those that exist.
2. This is known because there was an output of rows for that table when imputed to mySQL, also this is common knowledge for programmers.

Quiz Score: **34** out of 40

This quiz score has been manually adjusted by +1.0 points.