

Final Review: CS445

◆Review your homework, quizzes, previous exams, previous review sheets, and any tea leaves or used coffee grounds you can find. READ THE BOOK

Indexing

When optimizing a query, what operation are we trying to minimize?

An index file matches an index to a rid. What is an rid?

How are Type 1 and Type 2 index files different? How does this change the expense of accessing data via each type of index?

How is a hash function for a hash table or index lookup different than a hash function for password hashing?

We said in class that a B+ tree in a database index might have a fan-out as high as 100. What is fan-out for a tree? Why does a large fan-out benefit a database index? Be very specific in describing how a large fan out can reduce the cost of a query.

Why is it useful for the database catalog to track the number of pages used by a data file or the number of pages used by an index?

What is a covering index? Build a table and a query that demonstrates a covering index.

What are the benefits of using a hash index? What are the downsides?

What are the benefits of using a Tree index? What are the downsides?

If you have an index to support a foreign key lookup (for example, Table A has a foreign key that references a column in Table B. The index we are concerned with would be for Table B), do you think a hash or tree would be more effective? Defend your answer and be sure to include likely use cases.

Index nested loops join can be much more expensive than sort-merge join (used by Oracle) or a hash-join (used by MariaDB), neither of which rely on an index. Why would you ever build an index and do an index nested loops join?

The block nested loops join loads all or most of the smaller table into memory. Why does this help speed up the query?

Given the following relations with the given indexes [and index types]:

R1(a, b, c) Primary Key (a), Index (b) [Hash], Index(c) [Tree]

R2(x, y, z) Primary Key (x), Index (y, z) [Hash], Index(z) [Tree]

Which indexes, if any, could get used for each query?

select a from R1 where c > 100;

select a from R1 where b = 100;

select a, x from R1, R2 where b = y and z = 1;

select a, x from R1, R2 where y > c;

Relational Algebra

Build an expression to solve SQL Query #1, #2, and #3 under the **SQL queries** heading on the next page.

How much of SQL Query #6 below can be represented in Relational Algebra? What parts of the query can you not represent with Relational Algebra? Why do you think we don't have a relational operator for these parts?

SQLite (BONUS)

How is it different from MySQL

Give a use case where SQLite is more appropriate than MySQL/MariaDB

Web/DB architecture

Describe the input and output of each of the following pieces of the web architecture our project uses: Browser/Web server/PHP/Database.

Define LAMP.

Normalization/Redundancy

Define Functional Dependency.

List Armstrong's Axioms

Give an example of a trivial FD

List the requirements of 3NF, BCNF

Give an example of a table not in 1NF

Construct a relation, a set of functional dependencies, and a decomposition of said relation to demonstrate a **non-functional dependency preserving** decomposition. Give a brief explanation as to why this decomposition is **non-functional dependency preserving**.

List all the candidate keys in the following relations:

R { A, B, C, D } FD: $A \rightarrow B$ $B \rightarrow D$ $C \rightarrow D$

R { A, B, C, D, E } FD: $B \rightarrow AD$ $C \rightarrow AB$ $A \rightarrow CE$

R { A, B, C, D } FD: $A \rightarrow B$ $C \rightarrow D$ $B \rightarrow A$

List three super keys for each relation above.

Put each relation above into BCNF if it is not already. Is the result a lossless decomposition? Is the result a functional dependency preserving decomposition? Is the result a lossless decomposition?

Describe and give a concrete example of **Deletion Anomalies**, **Update Anomalies**, and **Insert Anomalies**. Build tables and queries to help explain your answer.

E-R diagrams

Explain why telephone number a bad primary key for a Person entity

Given the attached ER diagram, identify each Entity or Relation that would not be directly translated to a database Table. For each such relation or entity, explain where that object's data would live in the database.

Given the attached ER diagram, build the Create Table statements. Use your best guess as to appropriate data types.

Given the attached Create Table statements, build the ER diagram.

Given the attached ER diagram, can you track the price fluctuation in Corn over the last 10 years? If so, write a query that will produce the price of Corn for each year between 2007-2017. Otherwise, edit the ER diagram to allow you to track this information, then write the query.

SQL queries

Using your assignment one database:

1. Find all software that directly depends on **mauris id**. [Software Product Name]
2. Find all software that **mauris id** directly depends on [SoftwareID]
3. Find all software that **mauris id** indirectly depends on
(Exactly two steps away, **mauris id** -> X -> Y [only return Y]). [SoftwareID]
4. Find all software that is not depended on by other software. [SoftwareID]
5. Find every employee that is not a PointOfContact for any Client/Software pair. [EmpID]
6. Find the one employee that works on the most software projects (in the case of a tie, show the employee with the lowest EmpID) [EmpID]

Use the attached ER Diagram:

7. Find the CropManager with the highest PayBonus. [EmpID]
8. Find the CropManager with the highest (PayBonus + Wage). [FName, LName]
9. Find all the Crops with a below average Co-op Price [name]
10. Do any fields not have a worker trained on a TrackedTractor?
11. How many fields have a worker trained on a CombineHarvester [integer]?
12. Which employees were trained before August 28, 2017 [FName, LName]?
13. Which employee has the longest time between their first training and their last training [FName, LName]?

14. Write your own question that [requires you to write | is easier to write with] a subquery.

DBMS Features

List the features of a DBMS.

Define and give an example of each: schema, relation, entity, key constraint, descriptive attribute, candidate key, primary key, super key, ACID, Semantic Data Model, Relational Data Model, One to Many, many to many, one to one

Networking

Why do we use an SSH tunnel to connect to the database machine? Why is the tunnel *necessary* from your apartment? Why is the tunnel *easier* when sitting in the Scott Lab/CS Lab?

Explain the meaning of each of the parts of the ssh tunnel command:

```
ssh -L 3306:64.59.233.245:3306 zeus.cs.pacificu.edu  
-L  
3306:  
64.59.233.245:3306  
zeus.cs.pacificu.edu
```

FinalReview_ER_From_CT.sql

```
DROP TABLE IF EXISTS D;  
DROP TABLE IF EXISTS R1;  
DROP TABLE IF EXISTS A;  
DROP TABLE IF EXISTS C;  
DROP TABLE IF EXISTS B;
```

```
Create table B (  
    BID INT NOT NULL AUTO_INCREMENT PRIMARY KEY  
);
```

```
Create table C (  
    CID INT NOT NULL AUTO_INCREMENT PRIMARY KEY  
);
```

```
Create table A (  
    AID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    CID INT NOT NULL,  
    CONSTRAINT A_CID_FK FOREIGN KEY (CID) REFERENCES C(CID)  
);
```

```
Create table R1(  
    AID INT NOT NULL,  
    BID INT NOT NULL,  
    CONSTRAINT R1_AIDBID_PK PRIMARY KEY(AID, BID),  
    CONSTRAINT R1_AID_FK FOREIGN KEY (AID) REFERENCES A(AID),  
    CONSTRAINT R1_BID_FK FOREIGN KEY (BID) REFERENCES B(BID)  
);
```

```
Create table D (  
    DID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    AID INT NOT NULL,  
    BID INT NOT NULL,  
    CID INT NOT NULL,  
    CONSTRAINT D_AID_BID_FK FOREIGN KEY (AID, BID) REFERENCES R1(AID, BID),  
    CONSTRAINT D_CID_FK FOREIGN KEY (CID) REFERENCES C(CID),  
    CONSTRAINT D_CID_U UNIQUE(CID)  
);
```



