Teaching Advanced Topics in an Introductory Database Class

Presenters

Suhair Amer, Ph.D. Professor Department of Computer Science Southeast Missouri State University <u>samer@semo.edu</u> Mario Guimaraes, Ph.D. Professor and Chairperson Department of Computer Science Southeast Missouri State University <u>mguimaraes@semo.edu</u>

Abstract

This workshop will discuss Teaching Advanced Database Topics in an Intro to Database Class. In addition to teaching Relational Models, Database Design, and SQL, our institution covers optional topics such as: database security, data warehouse and data mining, relational algebra, stored procedures and triggers, concurrency, performance, introduction to SQL programming techniques: JDBC & SQLJ, web database programming using PHP, NoSQL- MongoDB, DB Ethics. We will debate over exercises, tools to use, and have an open discussion on which topics and what depth they should/can be covered.

Workshop outline

Part I (Suhair Amer) Structure of a DB course

- General information and course structure
 - Introduction to SQL programming techniques: JDBC & SQLJ
 - Web database programming with PHP
 - NoSQL- MongoDB

10 minutes break

> Part II (Mario Guimaraes) Advanced Database Topics

- Relational algebra
- Performance
- Stored procedures and triggers
- Database security
- Concurrency
- Data warehouse and data mining
- Database ethics

Part I

General information and course structure

• Catalog Class Description:

Basic concepts of database systems. Topics include the Entity Relationship and relational database models, normal forms, SQL, and the design and implementation of application databases. (3)

• **Prerequisite (s):** CS265 (Programming data structures with Java) with a minimum grade of C.

• Course learning outcomes:

1.Students will write complex queries to obtain information from a database.

- 2.Students will design a database using the Entity-Relationship model, apply normal forms to create appropriate tables and implement the database.
- 3.Students will work with at least one commercial database management system.

Book/reference:

• Fundamentals of database systems. Ramez Elmasri & Shamkant Navathe. Seventh Edition. Pearson. ISBN-10: 0-13-397077-9_

Software:

• Instructions will be provided. The assignment will have detailed instructions of how to install and use the software.

Topics covered

Week (s)	Topics Covered
1	Introductions forum, assignment 0 and quiz 0
2	Databases and database users
3	Database system concepts and architecture
4	Data modeling using the ER model
5	The EER model
6	the relational data model and relational DB constraints
7	SQL
8	SQL
9	Relational DB design by ER- and EER- to – Relational mapping
10	Introduction to SQL programming techniques: JDBC & SQLJ
11	Web database programming with PHP
12	Optional: NoSQL- MongoDB or Database Ethics (students' choice)
13-15	Project
16	Finals week

Assessment and Grading

Activity	Points	%
Introductions form	5	0.5
Practice Assignment	5	0.5
Practice Quiz	5	0.5
10 discussions (each 5 points)	50	5
10 Assignments (each 45 points)	450	45
10 Quizzes (each 30 points)	300	30
Project	125	12.5
Final exam	60	6

Letter grades assigned approximately on a 90, 80, 70, 60 scales. Your final score should be

at least 90% (900 points) to get an A at least 80% (800 points) to get a B at least 70% (700 points) to get a C at least 60% (600 points) to get a D Otherwise F

Mapping course objectives to activities

Course Objectives	Units/Topics	Activities
 A. Students will write complex queries to obtain information from a database. 	Chapters 7, 9-11 & supplemental material	Assignments 7-10 Quizzes 7-10 Discussion forums 7 -10 Project
B. Students will design a database using the Entity- Relationship model, apply normal forms to create appropriate tables and implement the database.	Chapters 1-7, 9-11 & supplemental material	Assignments 1-10 Quizzes 1-10 Discussion forums 1 -10 Project
C. Students will work with at least one commercial database management system.	Chapters 1-7, 9-11 & supplemental material	Project

Late work policies

Late work policy 1:

You can upload any late work to the late work drop box by 5:00 pm on the last Friday before finals week. I will not accept any activities submitted after that time. I don't accept activities submitted via email.

You need to include an excuse/reason.

Dr. Amer reserves the right to accept the excuse and grade the activities.

Excuses may include medical with documentation, family emergencies, military related, etc.

You can upload up to 5 activities.

You can only upload assignments, reports, projects, and discussions.

If you missed a quiz, you need to arrange to make it up with the faculty.

- > A lot of grading during final week and before submitting final grade
- Students complain that they did not know what their grade is before taking the final.
- Students with no emergencies took advantage of this policy.

Late work policy 2:

Please upload any late work (assignments, project reports, discussion) to the late work drop box by 11:59PM on the last Friday before finals week. No activities are accepted after that time and no activities are graded if sent via email.

Upload to the late work drop box.

You will loose 20% of the activity's grade no matter what the excuse is unless you get approval from faculty prior to missing the activity. You need to make arrangements to make up missed quizzes.

- Grading happened during final week and before submitting final grade.
- Mostly students with emergencies took advantage of this policy.

Late work policy 3:

Canvas marks your submitted/not submitted activity as:

- on time (no color)
- late (blue)
- missing (red)

The policy is that I don't accept late work. I reserve the right to not grade the activity if it is late. I will evaluate your excuse and decide accordingly. Late activities will be graded during finals week. Priority is given to grade activities submitted by the deadline. If I have time, I might grade them earlier.

A late penalty is applied to late work. 20% of score will be deducted from first 3 late activities. 40% from the next 3. 60% from the next three, etc. When completing an activity, please attach a note with your excuse.

You cannot make up / submit any work after the last Friday before finals.

I don't accept submissions via email.

Late work policy 4:

Canvas marks your submitted/not submitted activity (after the due date) as:

- on time (no color)
- late (blue)
- missing (red)
- •Priority is given to grade activities submitted on time.
- •Late activities are graded during finals (which is after the last day of the semester but before the final grade is submitted to the registrar)
- •The faculty reserves the right to not grade a late activity. You need to include your excuse in the comment section of the activity. Dr. Amer will check our excuse and grade it or not grade it accordingly.
- •You cannot submit/complete any work after the last day of classes.
- •Even if you submit your activity one minute late, it will still be marked by Canvas as late. If your activity is marked as late, it will be graded (if accepted by faculty) after the last day of classes.
- •Don't send emails regarding grading late activities.
- •A late penalty worth 50% of the activity's total score will be applied to all late work.

Structure

Week(s)	Activities
1	Introductions forum, practice assignment and practice quiz
2-12	 Cover 1 chapter per week By end of every week, students complete: Discussion forum Assignment Quiz
13-15	Project
Finals week	Final exam

∷ ▼ Unit 0 (Introductions)	Complete All Items 🔮 + 🗄
E Objectives 0 View	⊘ :
E Resources 0	⊘ :
ii 同 Hallway discussion	•
Activities to complete 0	•
Introductions forum Aug 23, 2021 5 pts	⊘ :
Practice assignment Aug 23, 2021 5 pts	⊘ :
Image: Second system Practice quiz Aug 23, 2021 5 pts	⊘ :
Reflective assignment (optional-not graded)	⊘ :

ii • Unit 2 (Chapter 2)	Complete All Items	+	:
ii Dijectives 2 View		•	:
⋮		Ø	:
O Chapter 2 slides O		⊘	:
ii 🗇 Hallway discussion		I	:
E Activities to complete 2		0	:
Image: Biscussion 2 (Chapter 2) Sep 8, 2021 5 pts		•	:
Image: Sep 8, 2021 45 pts		•	:
Image: Weight of the system Quiz 2 (Chapter 2) Sep 8, 2021 30 pts 30 pts		Ø	:
B Reflective assignment (optional-not graded)		Ø	:

Put	blished 🔊 Edit 🚦
Hallway discussion All Sections	2 2
Hallway discussion to imitate student interaction outside classroom.	
Faculty and graduate assistant will not be involved in this discussion nor will they answer questions posted here.	
Search entries or author 🛛 🔊 🗈 🖸	\checkmark Subscribe
← Reply	

Reflective assignment (optional-not graded)



Points	None		
Submitting	Nothing		
Due	For	Available from	Until
-	Everyone	-	
+ Rubric			

Previous

Next 🕨

Published

N Edit

Discussion forums

		Published 🗞 Edit 🗄
This is a graded discussion: 5	points possible	due Sep 11, 2022
Discussion 2 (Chapter	2)	240 240
Briefly explain or elaborate	e on any topic you found interesting in the chapters covered in this module.	
Make sure that you are dis	cussing something that was not covered by another student who previously posted to this discussion forum.	
Search entries or author	Unread © It	√ Subscribe
← Reply		

1 2 3 >>

In this chapter, I learned about the main concepts used in database systems and distinguished three main categories. high level or conceptual data models.

Low-level or physical data models and representational or implementation data models. The model does not change often, while the database state changes each time data is inserted, deleted, or modified. The topic that interests me more is data independence, which is very important in the following ways: 1. helps you improve the quality of your data. 2. database system maintenance becomes affordable. 3. enforces standards and improves database security. 4. you don't need to change the data structure in your application. 5. allows developers to focus on the overall structure of the database rather than worrying about the internal implementation . 6. It allows you to improve unbroken or unpartitioned state. 7. Database inconsistency is greatly reduced. 8. It is easy to make changes at the physical level, which is needed to improve system performance.

6 Reply

0

Issues encountered

- First: I would cover all material and then dedicate last 4 weeks for tools. Assign students two projects:
 - Project 1: detailed explanation of how to download MySQL, create a DB, Download Java, connect to DB, create GUI/forms, generate reports, how to use PHP, connect to DB and create GUI and generate reports.
 - Project 2: Choose Java or PHP and develop a system.
 - Last month, students were busy with holidays and finals and most completed successfully.
 - ➢When covering SQL, few students indicated that it would be better if they had hands-on practice (mainly non-regular students who were already working).

- Second, required students to download MySQL (free because of limited resources) early in the semester and assignments included (when appropriate) programming component.
 - Last 2-3 weeks dedicated to developing a project.
 - >Majority of students completed the project successfully.
 - ➤ Was able to resolve download and setup problems early in the semester.
 - Student had hands-on practice throughout the semester.

Required assignments

Week (s)	Topics Covered/activities	
1	Introductions forum, assignment 0 and quiz 0	
2	Databases and database users	
3	Database system concepts and architecture	Assignmen
4	Data modeling using the ER model	that require
5	The EER model	p. 08. a.i.i.i
6	the relational data model and relational DB constraints	
7	SQL	
8	SQL	
9	Relational DB design by ER- and EER- to – Relational mapping	
10	Introduction to SQL programming techniques: JDBC & SQLJ	
11	Web database programming with PHP	
12	Optional: NoSQL- MongoDB or Database Ethics (students' choice)	
13-15	Project	
16	Finals week	

ts no ng

Week (s)	Topics Covered/activities	
1	Introductions forum, assignment 0 and quiz 0	
2	Databases and database users	
3	Database system concepts and architecture	
4	Data modeling using the ER model	
5	The EER model	
6	the relational data model and relational DB constraints	
7	SQL	۲
8	SQL	
9	Relational DB design by ER- and EER- to – Relational mapping	
10	Introduction to SQL programming techniques: JDBC & SQLJ	
11	Web database programming with PHP	
12	Optional: NoSQL- MongoDB or Database Ethics (students' choice)	
13-15	Project	
16	Finals week	

Assignment that require programming

Open week 7 pdf assignment document: MySQL

- 1. Explain in detail how to install MySQL and go through an example of creating a DB.
- 2. The document lists steps and include pictures.
- 3. Then student will create a new DB similar to the one in the example.

Week (s)	Topics Covered/activities	
1	Introductions forum, assignment 0 and quiz 0	
2	Databases and database users	
3	Database system concepts and architecture	
4	Data modeling using the ER model	
5	The EER model	
6	the relational data model and relational DB constraints	
7	SQL	Assig
8	SQL	that
9	Relational DB design by ER- and EER- to – Relational mapping	progra
10	Introduction to SQL programming techniques: JDBC & SQLJ	
11	Web database programming with PHP	
12	Optional: NoSQL- MongoDB or Database Ethics (students' choice)	
13-15	Project	
16	Finals week	

Assignment that require programming

Open week 8 pdf assignment document: More MySQL

- 1. Based on previous week assignment.
- 2. Then student will create a new DB similar to the one in the example with advanced SQL queries.

Week (s)	Topics Covered/activities
1	Introductions forum, assignment 0 and quiz 0
2	Databases and database users
3	Database system concepts and architecture
4	Data modeling using the ER model
5	The EER model
6	the relational data model and relational DB constraints
7	SQL
8	SQL
9	Relational DB design by ER- and EER- to – Relational mapping
10	Introduction to SQL programming techniques: JDBC & SQLJ
11	Web database programming with PHP
12	Optional: NoSQL- MongoDB or Database Ethics (students' choice)
13-15	Project
16	Finals week

Assignment that require no programming

Week (s)	Topics Covered/activities
1	Introductions forum, assignment 0 and quiz 0
2	Databases and database users
3	Database system concepts and architecture
4	Data modeling using the ER model
5	The EER model
6	the relational data model and relational DB constraints
7	SQL
8	SQL
9	Relational DB design by ER- and EER- to – Relational mapping
10	Introduction to SQL programming techniques: JDBC & SQLJ
11	Web database programming with PHP
12	Optional: NoSQL- MongoDB or Database Ethics (students' choice)
13-15	Project
16	Finals week

Assignment that require programming

Open week 10 pdf assignment document: Introduction to SQL programming techniques: JDBC & SQLJ

- 1. Explain in detail how to install NetBeans and go through an example of creating a DB and a GUI (forms/reports).
- 2. The document lists steps and include pictures.
- 3. Then student will create a new DB similar to the one in the example and is required to insert, delete, etc. through the interface.

Week (s)	Topics Covered/activities
1	Introductions forum, assignment 0 and quiz 0
2	Databases and database users
3	Database system concepts and architecture
4	Data modeling using the ER model
5	The EER model
6	the relational data model and relational DB constraints
7	SQL
8	SQL
9	Relational DB design by ER- and EER- to – Relational mapping
10	Introduction to SQL programming techniques: JDBC & SQLJ
11	Web database programming with PHP
12	Optional: NoSQL- MongoDB or Database Ethics (students' choice)
13-15	Project
16	Finals week

Assignment that require programming

Open week 11 pdf assignment document: Web database programming with PHP

- 1. Explain in detail how to install PHP and go through an example of creating a DB and a GUI (forms/reports).
- 2. The document lists steps and include pictures.
- 3. Then student will create a new DB similar to the one in the example and is required to insert, delete, etc. through the interface.

Week (s)	Topics Covered/activities
1	Introductions forum, assignment 0 and quiz 0
2	Databases and database users
3	Database system concepts and architecture
4	Data modeling using the ER model
5	The EER model
6	the relational data model and relational DB constraints
7	SQL
8	SQL
9	Relational DB design by ER- and EER- to – Relational mapping
10	Introduction to SQL programming techniques: JDBC & SQLJ
11	Web database programming with PHP
12	Optional: NoSQL- MongoDB or Database Ethics (students' choice)
13-15	Project
16	Finals week

Assignment that require programming

Open week 12 pdf assignment document: MongoDB

This is for honor students/special project.

- 1. Explain in detail how to install MongoDB and go through an example.
- 2. The document lists steps and include pictures.
- 3. Then student will develop a system.

Week (s)	Topics Covered/activities
1	Introductions forum, assignment 0 and quiz 0
2	Databases and database users
3	Database system concepts and architecture
4	Data modeling using the ER model
5	The EER model
6	the relational data model and relational DB constraints
7	SQL
8	SQL
9	Relational DB design by ER- and EER- to – Relational mapping
10	Introduction to SQL programming techniques: JDBC & SQLJ
11	Web database programming with PHP
12	Optional: NoSQL- MongoDB or Database Ethics (students' choice)
13-15	Project
16	Finals week

Students choose between 10, 11 or 12 to complete project

Examples of projects
Develop a Wholesale Management System Database

Your DB will keep track of medical equipment used in hospitals

Ventilators Cardiopulmonary bypass device Dialysis machine Infusion pumps LASIK surgical machine Medical lasers Consult 120 Urine Analyzer Urine Reagent Strips 10 Parameter **Consult Liquid Urine Control** Plastic urine containers, sterile or unsterile Conical centrifuge tube, 15 ml Microscope slides and 1 coverslip Clinical Centrifuge Flow cytometers Blood gas analyzers Electrolyte analyzers Differential counters Coagulation analyzers Slide strainers Magnetic resonance imaging (MRI)

Deliverables:

DEVELOP A LIBRARY MANAGEMENT SYSTEM DATABASE PROJECT A student and faculty can issue books. Different limits for the number of books a student and teacher can issue. Also, the number of days will be distinct in the case of students and teachers for issue any book. Each book will have different ID. Also, each book of the same name and same author (but the number of copies) will have different ID. Entry of all the book will be done, who issue that book and when and also duration. Detail of Fine(when the book is not returned at a time) is also stored.

Deliverables:

DEVELOP RESTAURANT MANAGEMENT DATABASE PROJECT The restaurant maintains the catalog for the list of food and beverage items that it provides. Apart from providing food facility at their own premises, the restaurant takes orders online through their site. Orders on the phone are also entertained. To deliver the orders, we have delivery boys. Each delivery boy is assigned to the specific area code. The delivery boy cannot deliver outside the area which is not assigned to the delivery boy (for every delivery boy there can be a single area assigned to that delivery boy). The customer record is maintained so that premium customer can be awarded discounts.

Deliverables:

Blood donation system database project

A system in which data of patients, donors, blood bank, doctors/nurses would be saved and managed keeping in mind how all of them have an interrelation with each other

Deliverables:

Healthcare emergency care database project

A system to store and manage information regarding patients visiting the Emergency Care 24x7. The system help register patients, update patient diagnostic and prescription info by nurse or doctor. Keep track of payments.

Deliverables:

Animal Hospital Management database project A system to help store and manage information and procedures that vets have to deal with every day and keep track of the patients and their owners.

Deliverables:

Epidemic management system database project A system to help store and manage a database that presents the list of people who got affected by a disaster or manmade calamity.

Deliverables:

Develop a system allowing a faculty member to keep track of:

classes information students' information store student activities maintain grades detect plagiarism. Mainly you are designing and developing a system like Canvas and TurnitIn. Due to time limitations, the system should at least allow the faculty member to maintain:

student information such as name, address class information such as class name, number, location, time, and list of students enrolled. activities such as activity name and number (assignment1, lab3, quiz4, exam2, project1, etc.), maximum points, due date, instructions. student grade book for each activity scores/grades submitted documents. At this point you can choose the simplest file type. One type is sufficient. Make sure that you will be able to access the content of the document and process it to detect plagiarism. Faculty should be able to:

add information delete information update information check the new uploaded files against already submitted files and check for plagiarism. retrieve information check who has missing assignments who has the highest score who has the lowest score. what is the average score. The plagiarism is a major component of the project. You can Google " Free Plagiarism Detection Tools" to get ideas.

you can use MySQL, Java, PHP or any tool / programming language you prefer.

Deliverables:

In addition to your complete code and user manual, submit a report that consists of:

Introduction, design decisions, implementation decisions, testing results.

Print screens of your MySQL database

Print screens of your GUI (step by step showing how to use your system) and what to exepect.

Print screens of reports generated

You are running your own store. You want to keep track of your customers' information, merchandise information, orders, bills, receipts, likings, and feedback, restock-ing your merchandise, etc. You are free to choose the name of your store and what you sell. Make sure that you do not share your idea with other students. Similarities will be flagged as plagiarism and both of you will get an F.

Your store should be unique and is about something you like or is passionate about.

You will use MySQL to create your DB. However, you will create two interfaces: one through Java and the other through PHP to experience the differences between both types.

Deliverables:

Open: Develop a Wholesale Management System Database.docx Student project 1.docx Student project 2.docx

Same project idea, two different implementations





We will have a 10 minutes break and then resume with part II

Part II

Part II – Advanced Database Topics

- Relational algebra
- Performance
- Stored procedures and triggers
- Database security
- Concurrency
- Data warehouse and data mining
- Database ethics

Introduce Relational Algebra w/ SQL (not before)

Codd's 8 original relational algebra operations

- 1) Projection
- 2) Restriction (Selection)
- 3) Multiplication (Cartesian Product)
- 4) Join
- 5) Union
- 6) Intersect
- 7) Minus
- 8) Division

• Assignments: match relational algebra to SQL match pseudo-code to SQL

Relational_Algebra	SQL	Procedural Pseudo-code
1) R = Customers WHERE city = 'Dallas'	SELECT * FROM Customers WHERE city = 'Dallas';	For c1 = first row of customers until c1 = last row of customers If c1.city = 'Dallas' Display c1.* End-If End-For
2) R = customers [cid, cname]	SELECT cid, cname FROM customers;	For c1 = first row of customers until c1 = last row of customers Display c1.cid, c1.cname End-For
3) R = customers x orders	SELECT * FROM Customers, Orders;	For c1 = first row of customers until c1 = last row of customers For o1 = first row of orders until o1 = last row of orders Display c1.*, o1.* End-For End-For
4) R = customers join orders	SELECT * FROM Customers, Orders WHERE Customers.cid=Orders.cid;	For c1 = first row of customers until c1 = last row of customers For o1 = first row of orders until o1 = last row of orders If (c1.cid = o1.cid) Display c1.*, o1.* (all except for o1.cid) End-For Fnd-For
5) R = Students UNION Alumni	SELECT * FROM Students UNION SELECT * FROM Alumni	For s1 = first row of students until s1 = last row of students Display s1 End-For For a1 = first row of alumni until a1 = last row of alumni Display a1 End-For
6) R = Students INTERSECT Alumni	SELECT * FROM students INTERSECT SELECT * FROM alumni;	For s1 = first row of students until s1 = last row of students For a1 = first row of alumni until a1 = last row of alumni If (s1.ssn = a1.ssn and s1.name = a1.name) Display s1.* // display ssn and name End-If End-For End-For
7) R = Students MINUS Alumni	SELECT * FROM students MINUS SELECT * FROM alumni:	For s1 = first row of students until s1 = last row of students For a1 = first row of alumni until a1 = last row of alumni If (s1.ssn <> a1.ssn or s1.name <> a1.name) Display s1.* // display ssn and name End-If

Relational_Algebra	SQL
8) R1=Orders [cid, pid] / Products[pid] R2 = R1 Join Customers R3 = R2 [cname]	SELECT cname FROM customer c WHERE NOT EXISTS (SELECT * FROM products p WHERE NOT EXISTS (SELECT * FROM orders o WHERE c.cid = o.cid and p.pid = o.pid));

Relational_Algebra	Procedural Pseudo-code
8) R1=Orders [cid, pid] / Products[pid] R2 = R1 Join Customers R3 = R2 [cname]	<pre>FOR c from row 1 TO Last of customers foundProductNotOrdered = false p = row 1 OF products WHILE (foundProductNotOrdered = False AND p <= last row of products) o1 = first Row of Orders WHILE ((c.cid <> o.cid OR p.pid <> o.pid) && (o <= last row of orders) o1 = next row of orders IF (o > last row of orders) foundProductNotOrdered = True ELSE p1 = next row of products IF (foundProductNotOrdered = False) print customer name</pre>

Relational Algebra

Window Help Exit	
Dbjective	🔚 Relational Algebra
What are the names of the employees that work in accounting?	R1=Departments where dept_name='Acct'; R2=Employees JOIN R1; R3=R2[Name];
SQL with Join Select Name from Employees e, Departments d Where e.Dept_ID = d.Dept_ID and Dept_Name='Acct.';	Emp_ID Name Salary Dept_ID 100011 John Smith 30000 001
Select Name from Employees e where Dept_ID in	100022 Ean They 45000 002 100033 Mary Black 75000 001 100044 Andy Wallace 55000 001
(select Dept_ID from Departments where Dept_Name='Acct.');	Dept_ID Dept_Name Location
SQL with Sub-select, EXISTS	001 Acct. bldg2 002 HR bldg1
Select Name from Employees e where exists (select * from Departments d where e.Dept_ID = d.Dept_ID and Dept_Name='Acct.');	Buttons Next Step Reset

Window Help Exit

Objective
Join: Obtain rows from Tables employees and R1 that have a common value for the field Proj_ID.
SQL with NOT IN
Select Name from Employees e Where Emp_ID not in (select Emp_ID from EmpProj where Proj_ID='X001';
SQL with MINUS
Select Name from Employees minus (select Name from Employees e, EmpProj ep where e.Emp_ID = ep.Emp_ID and Proj_ID='X001');

🔲 R1 - Temporary Table 🦉			
Emp_ID	Proj_ID		
1234	X001		

🔲 R2 - 1	Tempora	ry Table		
Emp_ID	Name	Salary	Proj	ID
1234	Smith	40000	X001	

🔲 Relational Algebra	
R1=EMPPROJ where	Proj_ID='X001';
R2=R1 JOIN Employe	ees;
R3=R2[Name];	
R4=Employee[Name]	1

R5=R4	minus	R3;
-------	-------	-----

Employees - Source Table				
Emp_ID	Name	Salary		
1234	Smith	40000		
5678	Jones	65000	-	



Projects - Source Table				
Proj_ID	P_Name	StartDate		
X001	Proj. X	1/04/99	٠	
Y002	Proj. Y	3/12/00	•	



Performance

- Create Index
- Rewrite SQL (query plans)
- Redefine Main memory structures (SGA in Oracle)
- Change the Block Size
- Materialized Views, Denormalization
- Export/Import (drop indexes): defragment
- Check Locks
- Separate data by category in proper tablespaces (filegroups)
- Partition Database (Sharding in MongoDB)
- Redundant Arrays of Inexpensive Disks (RAID)
- Redefining Client-Server Architecture
- Buy better Hardware

Performance - Indexing

- When to Index
- Understand default index of the Product
- Understand other other indexes Product has
- Understand advantages and disadvantages of each Index

Performance Assignment

- 1) Do SELECT * FROM Orders where cid= 'c01';
- 2) Use explain and analyze command to find out the execution plan used.
 - (TutorialsPoint)
 - MySQL ANALYZE TABLE Statement (tutorialspoint.com)
 - MySQL EXPLAIN Statement (tutorialspoint.com)
- Use set PROFILING to find the speed of the sql statement (<u>https://dev.mysql.com/doc/refman/8.0/en/show-profile.html</u>).
- 2) Create an INDEX on the orders table.
- 3) Repeat steps one to three.
- 4) Download weather or historic data (of your choice)
- 5) Import it to SQL
- 6) Repeat steps 1-5
- Deliverable: screenshots showing successful execution of each step

Result: Execution plan will show as a series of relational algebra statements

Stored Procedures, Stored Functions, Triggers

- Present examples
- Explain advantages of each
- Compare Stored Procedure to Triggers
- Assignment: simple copy and paste OR change a stored procedure written in one DBMS to another one.
- Project: apply a stored procedure and a trigger in a meaningful way

Stored Procedures - example

Introduction Example			
Create a Stored Procedure Call the Stored Procedure Reset			
Create a Stored Procedure	VEHICLE		
	vid	make	model
SQL> CREATE OK REPLACE PROCEDUKE INSERTITICK (v01	Nissan	Pathfinder
VIU IN VARCHAR, make IN VARCHAR	v02	Ford	Escort
modelTyne IN VARCHAR	v03	Dodge	Neon
maxWeight IN NUMBER	v12	Ford	F-150
caraoType IN VARCHAR)			
IS			
(this procedure inserts a row in the supertype table VEHICLE and the corresponding subtype table TRUCK)			
BEGIN			
INSERT INTO VEHICLE VALUES (vid, make, modelType);			
INSERT INTO TRUCK VALUES (vid, maxWeight, cargoType);			
END;	TRUCK		
	vid	maxWeight	cargoType
	v01	2000	rice
	v12	3000	corn
Call the Stored Procedure			
SQL> exec insertTruck ('12', 'Ford', 'F-150', 3000, 'corn');			

Trigger Example

Trigger Code	DEPARTMENT				
SQL > CREATE OR REPLACE TRIGGER UpdateDeptTotal	Code	Name		Chair	TotalFacult
AFTER INSERT ON Faculty	CSE	Computer Science	an 121-11-11	11 2	
FOR EACH ROW	IEE	Industrial Enginee	ring 121-33-33	33 2	
This trigger increases the TotalFaculty attribute in the	MAT	Mathematics	121-68-66	66 1	
DEPARTMENT table when a new faculty member is inserted					
BEGIN					
UPDATE DEPARTMENT SET TotalFaculty = TotalFaculty + 1					
WHERE DEPARTMENT.Code = :new.dept;	FACULTY				
END	Pid	Firstname	Lastname	Rank	Dept
1	121-11-1111	Michael	Sonebraker	Emeritus	CSE
	121-22-2222	Bill	Gates	Assistant	CSE
Activate Trigger	121-33-3333	William	shockley	Assistant	IEE
COLS EVER in a second and the second s	121-55-5555	Adelle	Soldberg	Associate	IEE
SQLP EXEC INSERFACUIQ 121-44-4444 (Alan (Kay (Associate (USE)	121-66-6666	Alan	ouring	Full	MATH
	121-44-4444	Alan	(av	Associate	CSE

ntroduction	Example

xt Step Reset

Trigger Code

SQL > CREATE OR REPLACE TRIGGER UpdateDeptTotal

AFTER INSERT ON Faculty

FOR EACH ROW

--This trigger increases the TotalFaculty attribute in the DEPARTMENT table when a new faculty member is inserted

BEGIN

UPDATE DEPARTMENT SET TotalFaculty = TotalFaculty + 1 WHERE DEPARTMENT.Code = :new.dept;

END

Activate Trigger

SQL> EXEC insertFacult('121-44-4444','Alan','Kay','Associate','CSE')

DEPARTMENT

Code	Name	Chair	TotalFaculty
CSE	Computer Science an	121-11-1111	3
EE	Industrial Engineering	121-33-3333	2
MAT	Mathematics	121-66-6666	1

FACULTY

Pid	Firstname	Lastname	Rank	Dept
121-11-1111	Michael	Stonebraker	Emeritus	CSE
121-22-2222	Bill	Gates	Assistant	CSE
121-33-3333	William	Shockley	Assistant	IEE
121-55-5555	Adelle	Goldberg	Associate	IEE
121-66-6666	Alan	Touring	Full	MATH
121-44-4444	Alan	Kay	Associate	CSE

CREATE or REPLACE TRIGGER IncreaseDiscount

AFTER INSERT on Orders FOR EACH ROW

BEGIN

UPDATE Customers SET discnt = discnt+.1

WHERE

Customers.cid=:new.cid; END;

What is this trigger doing ?

CUSTOMERS

cid	cname	city	discrt
c001	TipTop	Duluth '	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	\$.00
c006	ACME	Kyoto	0.00

REDDUCTO

	1	1		1
pid	pname	city	quantity	price
p01	comb	Dallas	111400	0.50
p02	brush	Newark	203000	0.50
p03	razor	Duluth	150600	1.00
p04	pen	Duluth	12,5300	1.00
p05	pencil	Dallas	221400	1.00
p06	folder	Dallas	123100	2.00
p07	case	Newark	100500	1.00

ORDERS

ordno	month	cid	aid	pid	qty	dollars	
1011	- jan	c001	a01	pQ1	1000	450.00	
1012	jan	c001	a01	p01.	1000	450.00	
1019	feb	c001.	#02	p02	400	180.00	
1017	feb	c001	:06	p03	600	540.00	
1018	feb	c001	#03	p04	600	540.00	
1023	mar	c001	a04	p05	500	450.00	
1022	mar	c001	:05	p06	400	720.00	
1025	apr	c001	a05	p07	- 800	72.0.00	
1013	jan	c002	±03	p03	1000	880.00	
1026	may	c002	#05	p03	800	704.00	
1015	jan	c003_	a03	p05	1200	1104.00	
1014	jan	c003	a03	p05	1200	1104.00	
1021	leb	c004	a06	p01	1000	460.00	
1016	jan	c006	:01	p01	1000	500.00	
1020	feb	c006	±03	p07	600	600.00	
102.4	mar	c006	a06	p01	800	400.00	

AGENTS

aid	aname	city	percent
s01	Smith	New York	6
a02	Jones	Newark	6
±03	Brown	Tokyo	7
=04	Gray	New York	6
a05	Orasi	Duluth	5
a06	Smith	Dellas	5

1

Database Security Topics

DB Security Plan Database Access Control DB Application Threats: SQL injection, Inference Virtual Private Databases Encryption Auditing

DB Security Plan

• Exercise: search for three existing Database Security Plan (web, textbook, companies). Describe common points between them and difference between them.

Database Design

- ER to Tables
- Functional Dependencies -> Normalization

All non-key attributes are dependent on the key (1NF) the whole key (2NF) and nothing but the key (3NF)

• Induction (open a spreadsheet and put examples)

DB Access Control

- Student's login as DBA
- Create two different user accounts
- Login and have one user access another user account
- Grant Access
- Try the operation again

OR

Have users access each other's accounts (from the same group)

Grant each other access

Have them try again

DB Access Control

- Default Users and Passwords
 - Users, Passwords
 - Default users/passwords
 - sys, system accounts privileged, change default password
 - Sa (MS-SQL Server)
 - scott account well-known account/password, change it
 - general password policies (length, domain, changing, protection)
- People Having too many privileges
 - Privileges, Roles, Grant/Revoke
 - Privileges
 - System actions
 - Objects data
 - Roles (pre-defined and user-defined role)
 - Collections of system privileges (example: DBA role)
 - Grant / Revoke
 - Giving (removing) privileges or roles to (from) users

SQL Injection

- 1) Explain what SQL injection is
- 2) Present OWASP's ten threats
- 3) Use one of three alternatives depending on time constraints, place you are in.
- 3.1) Present the Solutions on SQL OWASP web-sites (intro do db class) OR
 - 3.2) Test for SQL injection on a simulation site (in a db security class) OR
 - 3.3) Test SQL injection on a real database (not recommended in US)

SQL Injection

- Web Application where
- 1) User inputs text into a textbox
- 2) Text used to build SQL Query dynamically
- 3) Malicious input changes the nature of the query

Example: user inputs names into textbox Name: **Benjamin Franklin ' OR 1=1;** SqlStr = "SELECT * FROM EMPLOYEE WHERE EMPLOYEE.Name = " + userInput

SqlStr = SELECT * FROM EMPLOYEE WHERE EMPLOYEE.Name = 'Benjamin Franklin' OR 1 = 1;

•Result: every row of the EMPLOYEE table will be returned

Virtual Private Databases (VPD)

- One Physical Database, many logical databases
- May be implemented through Views in most Database Management Systems (DBMS)

OR

- In Oracle, you may create a policy that will be fired when an operation (Insert, Update, etc.) is performed by a certain user on a certain object.
- Two users execute the same select statement and get different results because a WHERE clause is appended.
- Like SQL injection, SQL statement is modified at run-time (protect instead of hack)

VPD

- Recommended after SQL injection due to similar characteristics
- Students create a VPD with Views
- Three different users use the same SELECT statement and get different results.
- Use hierarchical and non-hierarchical access control
- Students explain how VPD is the same as SQL injection and how is it the opposite.
Virtual Private Database – w/ Oracle Policy



VPD, Example of Row Level Security w/ Views

stu	dents					6	Input			
	StudentID	LastName	Major	Status	GPA		mput			
•	8746	Smith	History	JR	3.00					
	7629	Knight	Accounting	SR	2.70					
	2637	Brown	Math	SR	3.50	Username:		lones ×		
	4856	Jones	CS	FR	2.80			volles		
	5347	Garcia	CS	SO	3.25					
Stu CR as	Ident_View EATE OR RE	EPLACE VIEW	/ Student_Viev	v				Next		

DATA.Student_View								
	StudentID	LastName	Major	Status	GPA			
	4856	Jones	CS	FR	2.80			

Encryption

- Encrypting Data at Rest versus Data in Transit (client-server)
- Encrypting at the OS Level versus Database Level versus Application Level

In Intro class: students explain advantages of encrypting at each level Students presented to how you can encrypt at each level

In da Database Security class: have students encrypt

Database Auditing

• Students copy and paste code to audit database

Or

Students modify code from one db (MySQL, Oracle, MS-SQL server) to another

Concurrency - Overview

- Shared versus Exclusive
- Unit of Locking (Field, Row, Table, DB)
- Explicit versus Implicit
- Releasing Locks
- Wait versus NoWait

Animations

http://adbc.kennesaw.edu -> transactions -> concurrency

Deadlock



Simple Transaction

🚳 oncurrency - Database E	Example			
Without Record Locking	With Record L	ocking Deadlock	Serial Locking	
Accounts				
AcctNo Balance	e Tyj	pe SSN		
10051 2000.00	C	222.11.1111 111.11.1111		
10053 500.00	c	111.33.1111		
10054 1200.00	s	111.11.1111		
2000.00	5	222.11.1111		
Transactions				
Transaction 1 A:Update accounts Set balance=balan Where accountNo	s ice-1500 =10052;	Transaction 2 B:Select balance into From accounts Where accountNo=1 C:Select balance into From accounts Where accountNo=1 D:If (v_balance1+v_b Print "Deny Credit Ca	o :v_balance1 0052; o :v_balance2 0054; alance2<2000) ird'';	10054 (fourth row) will have its balance changed from 1200.00 to 2700.00.
Set balance=balan	ce+1500	Eise Print "Issue Credit Ca	ard";	
Where accountNo	=10054;			
v_balance1 v_b 100.00 120	0.00			
		🛅 Stat	us	
		Deny Cr	edit Card	

Database Warehouse

- Presented to a DW architecture
- Star versus 3NF
- Students do an ETL with Python (stock market or weather data)
- Students see a video of an ETL with MS-SQL server
- Compare ETL with Python versus MS-SQL server or another tool

Example of Data Warehouse



Data Warehouse - Typical Daily Operations

• OLTP

Insert

Update

Delete

Select

Data Warehouse

Inserts in batch

Select retrieving many records

Data Warehouse

- History (Inmon, Kimball)
- Intro to Data Warehouse
- <u>https://www.youtube.com/watch?v=zTs5zjSXnvs</u>

Data Warehouse Toolkit – Kimball

- SSIS (soft to do ETL) Wiseowl tutorials
- <u>https://www.youtube.com/watch?v=3cPq9FXk-RA</u>

ETL may be done with tool such as SSIS or programming language such as Python. Staging area: typically EXCEL.

Data Warehouse – Stock Market, Daily versus Intra-day



• Identify Daily, Intra-day

DW Design (star) versus OLTP Design (3NF)



- Identify Fact, Dimension
- Identify Advantage Fact, Dimension
- Speed of Queries (DW) versus Integrity of Updates (OLTP)

Data Mining

- Explain Data Mining
- Use tutorialspoint. Explain induction
- Show examples of solutions of data Mining with Python versus MS-SQL Server

Data Mining: a very simple example of pattern recognition

- Data Mining: involves induction
- Gauss formula for numbers from 1 to N

$$S = \frac{n(n+1)}{2}$$

1+99, 2+98, 3+97, 4+96 ...

Ethics

• Awareness

Present a few scenarios

- Compare Data Mining with Discrimination
- Questions:

1) Describe one thing you learned about ethics ?

2) How is Data Mining similar to Discrimination ?

Project

Deliverables

0. Form a Team & Topic

1. Elaborate a topic and Summary (Problem Description. Click <u>here</u> for an example.

2. Define the Client. Who (client) will use the database ?

3. Create Sample Questions that your database will answer. Note that in your labs you were required to turn in the a)

question, b) the c) sql question. At this stage, I want only the questions in English.

Other Examples:

List name of films displaying at Capital Mall on Wednesday.

List times that a certain film still has seats available.

Draw an E-R Diagram – 1st draft

Conclude the E-R diagram

Relational Schema with keys indicated (E-R + FK) and Data Dictionary

CREATE TABLES (sql)

INSERT ROWS (sql)

List containing Forms

CRUD Matrix (Forms versus Tables)

Queries (state the query, the SQL statement and the Result) + Printout of Sample Forms and Reports with Data, 1st draft

Enhance Queries

Trigger, Stored Procedure and Functions (apply in a meaningful way)

Connect w/ Front-end – Web

Additional database administration features discussed in class or that you research (security concerns addressed)

Presentation

Project Deliverable (unique)

ACCESS MATRIX

Tables x	Categories	Customers	Employees	Orders	Orders	Suppliers	Products
Forms				Details			
Categories	CRUD						
Customer		R					
Labels							
Customers		CRUD					
Orders		R	R	CRUD	CRUD		RU
Employees			CRUD				
Suppliers						CRUD	
Products							CRUD

C = Create or Insert a Record (row)

R = Read/Query or Select a row

U = Update or Modify

D = Delete

Which is the most complex program/form to implement?

If a data type is changed in the Customers table, what programs/forms may need modification?

The Orders form accesses how many tables? The Employees table is accessed by how many programs/forms?

Thank You!