HW3A

May 15, 2025

W4111_2025_002_1: Introduction to Databases:Homework 3A

1 Overview

1.1 Scope

The material in scope for this homework is: - The content of lectures: - This includes all of the meterial covered in the lectures. - All of the material in the slides from lectures 1, 2, and 3. Students should read slides not covered in the lecture. - Lecture 4 slides except for the slides after slide 88. - Lecture 5 slides from slide 1 to slide 57. - The slides associated with the recommended textbook for - Chapter 1. - Chapter 2. - Chapter 3. - Chapter 4 slides 4.4 to 4.52 except for slide 4.35 (Transactions).

1.2 Submission Instructions

- Due date: 2025-Mar-09, 11:59 PM EDT on GradeScope.
- You submit on GradeScope. We will create a GradeScope submission for the homework.
- Your submission is a PDF of this notebook. You must tag the submission with locations in the PDF for each question. You must solve problems you experience producing a PDF including images. Please do not wait until the last minute.

There is a post/mega-thread on Ed Discussions that we will use to resolve questions and issues with respect to homework 3.

1.3 Brevity

Brevity

Students sometimes just write a lot of words hoping to get something right. We will deduct points if your answer is too long.

2 Initialization

2.1 Python Environment

[1]:
```
import copy
```

[1]:
```
import json
```

[2]:
```
import pandas
```

[4]:
```
# You should have installed the packages for previous homework assignments
#
import pymysql
import sqlalchemy
from sqlalchemy import create_engine
```

[5]:
```
import numpy
```

[3]:
```
# You have installed and configured ipython-sql for previous assignments.
# https://pypi.org/project/ipython-sql/
#
%load_ext sql
```

[4]:
```
# This is a hack to fix a version problem/incompatibility  with some of the␣
 ↪packages and magics.
#
%config SqlMagic.style = '_DEPRECATED_DEFAULT'
```

[5]:
```
# Make sure that you set these values to the correct values for your␣
 ↪installation and
# configuration of MySQL
#
db_user = "root"
db_password = "rootpass"
```

[6]:
```
# Create the URL for connecting to the database.
# Do not worry about the local_infile=1, I did that for wizard reasons that you␣
 ↪should not have to use.
#
db_url = f"mysql+pymysql://{db_user}:{db_password}@localhost?local_infile=1"
```

[7]:
```
# Initialize ipython-sql
#
%sql $db_url
```

[8]:
```
# Your answer will be different based on the databases that you have created on␣
 ↪your local MySQL instance.
#
```

```
%sql show tables from db_book
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
11 rows affected.
```

```
[8]: [('advisor',),
      ('classroom',),
      ('course',),
      ('department',),
      ('instructor',),
      ('prereq',),
      ('section',),
      ('student',),
      ('takes',),
      ('teaches',),
      ('time_slot',)]
```

```
[9]: from sqlalchemy import create_engine
     default_engine = create_engine("mysql+pymysql://root:rootpass@localhost/db_book?
       ↪local_infile=1")
     default_engine = default_engine.connect()
```

```
[10]: from sqlalchemy import inspect
      # Use the inspect module to get information about the engine
      inspector = inspect(default_engine)

      # Get the list of table names
      tables = inspector.get_table_names()

      # Display the tables
      tables_df = pandas.DataFrame(tables, columns=["Tables in db_book"])
      tables_df
```

```
[10]:    Tables in db_book
      0            advisor
      1          classroom
      2             course
      3         department
      4         instructor
      5             prereq
      6            section
      7            student
      8              takes
      9            teaches
      10         time_slot
```

2.2 Load Classic Models Dataset

You must load the `classicmodels` MySQL sample database.

You can use DataGrip and follow the process/tasks you did in HW 0 to load `db_book`. The page describing the dataset also links a tutorial on how to load the dataset. The installation file is a set of SQL DDL and DML statements. You ran a file with similar statements for HW 0.

The following query tests correct loading.

[11]:

```
%%sql

use classicmodels;

with one as (select customerNumber,
                    (select customerName
                     from customers
                     where customers.customerNumber = orders.customerNumber) as␣
 ↪customerName,
                 orderNumber
             from orders
             where (select country from customers where customers.
 ↪customerNumber = orders.customerNumber) = 'Spain'),
two as (
    select
        *
    from one join orderdetails using(orderNumber)
)
select
    customerNumber, customerName,
    count(*) as noOfOrders,
    concat("$", format(sum(quantityOrdered*priceEach), 2)) as revenue
from two
group by customerNumber, customerName
having sum(quantityOrdered*priceEach) >= 50000
order by sum(quantityOrdered*priceEach) desc;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
0 rows affected.
4 rows affected.
```

[11]:

```
[(141, 'Euro+ Shopping Channel', 259, '$820,689.54'),
 (458, 'Corrida Auto Replicas, Ltd', 32, '$112,440.09'),
 (216, 'Enaco Distributors', 23, '$68,520.47'),
 (484, 'Iberia Gift Imports, Corp.', 15, '$50,987.85')]
```

3 Written Questions

3.1 IsA

Question

Briefly explain the concepts of: - Inheritance/specialization/generalization. - Incomplete/complete. - Disjoint/overlapping.

Answer

Inheritence, specialization, and generalizations are extended E-R features.

- **Specialization** is "the process of designating subgroupings within an entity set" (Silberschatz, 272). We view this as a "top-down design process," since distinctions are made to specificy certain subgroups that allows DBMs/Ontologists/users to better disguish entities (as well as "extend" additional attributes onto them, *see below*).

In diagramming, we should designate specialization with a "hollow arrow-head pointing" from the specizalized (hyponymic) entity to the other (hypernymic) entity (Silberschatz et al., 272). This is a way of designating the IS-A relationship.

- **Overlapping specialization** would mean that an entity "may belong to multiple specialized entity sets" that extend their attriubutes, while **disjoint specialization** allows at most one specialized entity set.
- Generalization is containment relation, usually discovered through a bottom-up design proceess, w entity sets might synthesize into a higher-level entity set "on the basis of commmon features" (273).

What makes these concepts of specialization and generalization particularly useful is **attribute inheritence**.

On both generalization/specialization, there is a **completeness constraint**, which "specifies whether or not an entity in the higher0level entity set must belong to at least one of the lower-level entity sets within the generalization/specilization" (275). **Total specialization/generalization** means that "each higher-level entity must belong to a lower-level entity" (ibid.) In other words, every employee must be designated further by a subgrouping of what kind of employee they are. **Partial specialization/generalization**, on the other hand, means that some entities may not belong to any lower-level entity set, which means you could be an employee without any further specification.

3.2 Views

Question

Views are a powerful concept in relational databases, and databases in general. Succinctly give 3 motivations for/reasons to use views.

Answer

1. Abstraction and simplification - Views provide a simplified interface to complex queries. They can hide joins, filters, or subqueries, presenting less familiar database users with a clean, logical representation of the data.

2. Security - Views can restrict access to sensitive columns or rows by exposing only selected fields or filtered data. You can grant permissions on views instead of base tables.
3. Independence/Consistency on Application side - Views allow the underlying schema to change (e.g., table splits or joins) while keeping applications and users insulated from those changes. For example, if you split a table, a view can “preserve” the older schema, which will allow users to stay familiar with the data/query, even while the underlying schema is change.

3.3 Types of Views

Question

Consider the following SQL statement that creates a view.

```
with one as (select customerNumber,
                    (select customerName
                     from customers
                     where customers.customerNumber = orders.customerNumber) as customerName,
                 orderNumber
             from orders),
two as (
    select
        *
    from one join orderdetails using(orderNumber)
)
select
    customerNumber, customerName,
    count(*) as noOfOrders,
    concat("$", format(sum(quantityOrdered*priceEach), 2)) as revenue
from two
group by customerNumber, customerName
having sum(quantityOrdered*priceEach) >= 50000
order by sum(quantityOrdered*priceEach) desc;
```

Assume that: 1. The total number of orders over all customers is very large. 2. There are relatively few customers. 3. Many users frequently query (read) the view. 4. The rate of updating an order or creating and order is small.

What type of view would you use and why?

Answer

Here is where a materialized view comes in handy because unlike a regular view (which recomputes the result every time it’s queried), a materialized view stores the result set physically on disk. The fact that the result is pre-computed usually will result in users getting faster response times from stored results, and the more users there are reading/querying the view, the greater the “ROI” for materializing is. Because we assume that there is a small rate of updating, a materialized view will work well here, since they are mostly ideal for aggregations, joins, and expensive queries that don’t need real-time accuracy.

3.4 View Expansion

Question

Consider the following SQL statements:

```
create or replace view hw3_customerSummary as select
    customerNumber, customerName, contactFirstName, contactLastName, phone as contactPhone, sal
from
    customers;

create or replace view hw3_customersAndSales as
select
    customerNumber, customerName, contactFirstName, contactLastName, contactPhone,
    employeeNumber as salesRepEmploueeNumber,
    concat(employees.lastName, ",", employees.firstName) as salesRepEmployeeName,
    email as saleRepEmployeeEmail
from
    hw3_customerSummary join employees on employeeNumber=salesRepEmployeeNumber;
```

Show the resulting query after view expansion for the following query.

```
select * from hw3_customersAndSales;
```

Answer

SELECT customers.customerNumber, customers.customerName, customers.contactFirstName, customers.contactLastName, customers.phone AS contactPhone, employees.employeeNumber AS salesRepEmployeeNumber, CONCAT(employees.lastName, “,”, employees.firstName) AS salesRepEmployeeName, employees.email AS saleRepEmployeeEmail FROM customers JOIN employees ON employees.employeeNumber = customers.salesRepEmployeeNumber;

[12]:

```
%%sql
create or replace view hw3_customerSummary as select
    customerNumber, customerName, contactFirstName, contactLastName, phone as␣
  ↪contactPhone, salesRepEmployeeNumber
from
    customers;

create or replace view hw3_customersAndSales as
select
    customerNumber, customerName, contactFirstName, contactLastName,␣
  ↪contactPhone,
    employeeNumber as salesRepEmploueeNumber,
    concat(employees.lastName, ",", employees.firstName) as␣
  ↪salesRepEmployeeName,
    email as saleRepEmployeeEmail
from
    hw3_customerSummary join employees on employeeNumber=salesRepEmployeeNumber;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
```

```
0 rows affected.
0 rows affected.
```

[12]:

```
[]
```

[13]:

```
%%sql
select * from hw3_customersAndSales;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
100 rows affected.
```

[13]:

```
[(103, 'Atelier graphique', 'Carine ', 'Schmitt', '40.32.2555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (112, 'Signal Gift Stores', 'Jean', 'King', '7025551838', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (114, 'Australian Collectors, Co.', 'Peter', 'Ferguson', '03 9520 4555', 1611,
'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (119, 'La Rochelle Gifts', 'Janine ', 'Labrune', '40.67.8555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (121, 'Baane Mini Imports', 'Jonas ', 'Bergulfsen', '07-98 9555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (124, 'Mini Gifts Distributors Ltd.', 'Susan', 'Nelson', '4155551450', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (128, 'Blauer See Auto, Co.', 'Roland', 'Keitel', '+49 69 66 90 2555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (129, 'Mini Wheels Co.', 'Julie', 'Murphy', '6505555787', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (131, 'Land of Toys Inc.', 'Kwai', 'Lee', '2125557818', 1323, 'Vanauf,George',
'gvanauf@classicmodelcars.com'),
 (141, 'Euro+ Shopping Channel', 'Diego ', 'Freyre', '(91) 555 94 44', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (144, 'Volvo Model Replicas, Co', 'Christina ', 'Berglund', '0921-12 3555',
1504, 'Jones,Barry', 'bjones@classicmodelcars.com'),
 (145, 'Danish Wholesale Imports', 'Jytte ', 'Petersen', '31 12 3555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (146, 'Saveley & Henriot, Co.', 'Mary ', 'Saveley', '78.32.5555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (148, 'Dragon Souveniers, Ltd.', 'Eric', 'Natividad', '+65 221 7555', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (151, 'Muscle Machine Inc', 'Jeff', 'Young', '2125557413', 1286, 'Tseng,Foon
Yue', 'ftseng@classicmodelcars.com'),
 (157, 'Diecast Classics Inc.', 'Kelvin', 'Leong', '2155551555', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (161, 'Technics Stores Inc.', 'Juri', 'Hashimoto', '6505556809', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (166, 'Handji Gifts& Co', 'Wendy', 'Victorino', '+65 224 1555', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com'),
 (167, 'Herkku Gifts', 'Veysel', 'Oeztan', '+47 2267 3215', 1504, 'Jones,Barry',
'bjones@classicmodelcars.com'),
 (168, 'American Souvenirs Inc', 'Keith', 'Franco', '2035557845', 1286,
'Tseng,Foon Yue', 'ftseng@classicmodelcars.com'),
 (171, 'Daedalus Designs Imports', 'Martine ', 'Rancé', '20.16.1555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (172, "La Corne D'abondance, Co.", 'Marie', 'Bertrand', '(1) 42.34.2555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (173, 'Cambridge Collectables Co.', 'Jerry', 'Tseng', '6175555555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (175, 'Gift Depot Inc.', 'Julie', 'King', '2035552570', 1323, 'Vanauf,George',
'gvanauf@classicmodelcars.com'),
 (177, 'Osaka Souveniers Co.', 'Mory', 'Kentary', '+81 06 6342 5555', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (181, 'Vitachrome Inc.', 'Michael', 'Frick', '2125551500', 1286, 'Tseng,Foon
Yue', 'ftseng@classicmodelcars.com'),
 (186, 'Toys of Finland, Co.', 'Matti', 'Karttunen', '90-224 8555', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (187, 'AV Stores, Co.', 'Rachel', 'Ashworth', '(171) 555-1555', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (189, 'Clover Collections, Co.', 'Dean', 'Cassidy', '+353 1862 1555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (198, 'Auto-Moto Classics Inc.', 'Leslie', 'Taylor', '6175558428', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (201, 'UK Collectables, Ltd.', 'Elizabeth', 'Devon', '(171) 555-2282', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (202, 'Canadian Gift Exchange Network', 'Yoshi ', 'Tamuri', '(604) 555-3392',
1323, 'Vanauf,George', 'gvanauf@classicmodelcars.com'),
 (204, 'Online Mini Collectables', 'Miguel', 'Barajas', '6175557555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (205, 'Toys4GrownUps.com', 'Julie', 'Young', '6265557265', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (209, 'Mini Caravy', 'Frédérique ', 'Citeaux', '88.60.1555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (211, 'King Kong Collectables, Co.', 'Mike', 'Gao', '+852 2251 1555', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (216, 'Enaco Distributors', 'Eduardo ', 'Saavedra', '(93) 203 4555', 1702,
'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (219, 'Boards & Toys Co.', 'Mary', 'Young', '3105552373', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (227, 'Heintze Collectables', 'Palle', 'Ibsen', '86 21 3555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (233, 'Québec Home Shopping Network', 'Jean ', 'Fresnière', '(514) 555-8054',
1286, 'Tseng,Foon Yue', 'ftseng@classicmodelcars.com'),
 (239, 'Collectable Mini Designs Co.', 'Valarie', 'Thompson', '7605558146',
1166, 'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (240, 'giftsbymail.co.uk', 'Helen ', 'Bennett', '(198) 555-8888', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (242, 'Alpha Cognac', 'Annette ', 'Roulet', '61.77.6555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (249, 'Amica Models & Co.', 'Paolo ', 'Accorti', '011-4988555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (250, 'Lyon Souveniers', 'Daniel', 'Da Silva', '+33 1 46 62 7555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (256, 'Auto Associés & Cie.', 'Daniel ', 'Tonini', '30.59.8555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (259, 'Toms Spezialitäten, Ltd', 'Henriette ', 'Pfalzheim', '0221-5554327',
1504, 'Jones,Barry', 'bjones@classicmodelcars.com'),
 (260, 'Royal Canadian Collectables, Ltd.', 'Elizabeth ', 'Lincoln', '(604)
555-4555', 1323, 'Vanauf,George', 'gvanauf@classicmodelcars.com'),
 (276, "Anna's Decorations, Ltd", 'Anna', "O'Hara", '02 9936 8555', 1611,
'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (278, 'Rovelli Gifts', 'Giovanni ', 'Rovelli', '035-640555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (282, 'Souveniers And Things Co.', 'Adrian', 'Huxley', '+61 2 9495 8555', 1611,
'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (286, "Marta's Replicas Co.", 'Marta', 'Hernandez', '6175558555', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (298, 'Vida Sport, Ltd', 'Mihael', 'Holz', '0897-034555', 1702,
'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (299, 'Norway Gifts By Mail, Co.', 'Jan', 'Klaeboe', '+47 2212 1555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (311, 'Oulu Toy Supplies, Inc.', 'Pirkko', 'Koskitalo', '981-443655', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (314, 'Petit Auto', 'Catherine ', 'Dewey', '(02) 5554 67', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (319, 'Mini Classics', 'Steve', 'Frick', '9145554562', 1323, 'Vanauf,George',
'gvanauf@classicmodelcars.com'),
 (320, 'Mini Creations Ltd.', 'Wing', 'Huang', '5085559555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (321, 'Corporate Gift Ideas Co.', 'Julie', 'Brown', '6505551386', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (323, 'Down Under Souveniers, Inc', 'Mike', 'Graham', '+64 9 312 5555', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com'),
 (324, 'Stylish Desk Decors, Co.', 'Ann ', 'Brown', '(171) 555-0297', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (328, 'Tekni Collectables Inc.', 'William', 'Brown', '2015559350', 1323,
'Vanauf,George', 'gvanauf@classicmodelcars.com'),
 (333, 'Australian Gift Network, Co', 'Ben', 'Calaghan', '61-7-3844-6555', 1611,
'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (334, 'Suominen Souveniers', 'Kalle', 'Suominen', '+358 9 8045 555', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (339, 'Classic Gift Ideas, Inc', 'Francisca', 'Cervantes', '2155554695', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (344, 'CAF Imports', 'Jesus', 'Fernandez', '+34 913 728 555', 1702,
'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (347, "Men 'R' US Retailers, Ltd.", 'Brian', 'Chandler', '2155554369', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (350, 'Marseille Mini Autos', 'Laurence ', 'Lebihan', '91.24.4555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (353, 'Reims Collectables', 'Paul ', 'Henriot', '26.47.1555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (357, 'GiftsForHim.com', 'Wales', 'MacKinlay', '64-9-3763555', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com'),
 (362, 'Gifts4AllAges.com', 'Juri', 'Yoshido', '6175559555', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (363, 'Online Diecast Creations Co.', 'Dorothy', 'Young', '6035558647', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (376, 'Precious Collectables', 'Braun', 'Urs', '0452-076555', 1702,
'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (379, 'Collectables For Less Inc.', 'Allen', 'Nelson', '6175558555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (381, 'Royale Belge', 'Pascale ', 'Cartrain', '(071) 23 67 2555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (382, 'Salzburg Collectables', 'Georg ', 'Pipps', '6562-9555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (385, 'Cruz & Sons Co.', 'Arnold', 'Cruz', '+63 2 555 3587', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (386, "L'ordine Souveniers", 'Maurizio ', 'Moroni', '0522-556555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (398, 'Tokyo Collectables, Ltd', 'Akiko', 'Shimamura', '+81 3 3584 0555', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (406, 'Auto Canal+ Petit', 'Dominique', 'Perrier', '(1) 47.55.6555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (412, 'Extreme Desk Decorations, Ltd', 'Sarah', 'McRoy', '04 499 9555', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com'),
 (415, 'Bavarian Collectables Imports, Co.', 'Michael', 'Donnermeyer', ' +49 89
61 08 9555', 1504, 'Jones,Barry', 'bjones@classicmodelcars.com'),
 (424, 'Classic Legends Inc.', 'Maria', 'Hernandez', '2125558493', 1286,
'Tseng,Foon Yue', 'ftseng@classicmodelcars.com'),
 (447, 'Gift Ideas Corp.', 'Dan', 'Lewis', '2035554407', 1323, 'Vanauf,George',
'gvanauf@classicmodelcars.com'),
 (448, 'Scandinavian Gift Ideas', 'Martha', 'Larsson', '0695-34 6555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (450, 'The Sharp Gifts Warehouse', 'Sue', 'Frick', '4085553659', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (452, 'Mini Auto Werke', 'Roland ', 'Mendel', '7675-3555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (455, 'Super Scale Inc.', 'Leslie', 'Murphy', '2035559545', 1286, 'Tseng,Foon
Yue', 'ftseng@classicmodelcars.com'),
 (456, 'Microscale Inc.', 'Yu', 'Choi', '2125551957', 1286, 'Tseng,Foon Yue',
'ftseng@classicmodelcars.com'),
 (458, 'Corrida Auto Replicas, Ltd', 'Martín ', 'Sommer', '(91) 555 22 82',
1702, 'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (462, 'FunGiftIdeas.com', 'Violeta', 'Benitez', '5085552555', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (471, 'Australian Collectables, Ltd', 'Sean', 'Clenahan', '61-9-3844-6555',
1611, 'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (473, 'Frau da Collezione', 'Franco', 'Ricotti', '+39 022515555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (475, 'West Coast Collectables Co.', 'Steve', 'Thompson', '3105553722', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (484, 'Iberia Gift Imports, Corp.', 'José Pedro ', 'Roel', '(95) 555 82 82',
1702, 'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (486, 'Motor Mint Distributors Inc.', 'Rosa', 'Salazar', '2155559857', 1323,
'Vanauf,George', 'gvanauf@classicmodelcars.com'),
 (487, 'Signal Collectibles Ltd.', 'Sue', 'Taylor', '4155554312', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (489, 'Double Decker Gift Stores, Ltd', 'Thomas ', 'Smith', '(171) 555-7555',
1501, 'Bott,Larry', 'lbott@classicmodelcars.com'),
 (495, 'Diecast Collectables', 'Valarie', 'Franco', '6175552555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (496, "Kelly's Gift Shop", 'Tony', 'Snowden', '+64 9 5555500', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com')]
```

[14]:

```
%%sql
SELECT
    customers.customerNumber,
    customers.customerName,
    customers.contactFirstName,
    customers.contactLastName,
    customers.phone AS contactPhone,
    employees.employeeNumber AS salesRepEmployeeNumber,
    CONCAT(employees.lastName, ",", employees.firstName) AS␣
 ↪salesRepEmployeeName,
    employees.email AS saleRepEmployeeEmail
FROM
    customers
    JOIN employees ON employees.employeeNumber = customers.
 ↪salesRepEmployeeNumber;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
100 rows affected.
```

[14]:

```
[(103, 'Atelier graphique', 'Carine ', 'Schmitt', '40.32.2555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (112, 'Signal Gift Stores', 'Jean', 'King', '7025551838', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (114, 'Australian Collectors, Co.', 'Peter', 'Ferguson', '03 9520 4555', 1611,
'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (119, 'La Rochelle Gifts', 'Janine ', 'Labrune', '40.67.8555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (121, 'Baane Mini Imports', 'Jonas ', 'Bergulfsen', '07-98 9555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (124, 'Mini Gifts Distributors Ltd.', 'Susan', 'Nelson', '4155551450', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (128, 'Blauer See Auto, Co.', 'Roland', 'Keitel', '+49 69 66 90 2555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (129, 'Mini Wheels Co.', 'Julie', 'Murphy', '6505555787', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (131, 'Land of Toys Inc.', 'Kwai', 'Lee', '2125557818', 1323, 'Vanauf,George',
'gvanauf@classicmodelcars.com'),
 (141, 'Euro+ Shopping Channel', 'Diego ', 'Freyre', '(91) 555 94 44', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (144, 'Volvo Model Replicas, Co', 'Christina ', 'Berglund', '0921-12 3555',
1504, 'Jones,Barry', 'bjones@classicmodelcars.com'),
 (145, 'Danish Wholesale Imports', 'Jytte ', 'Petersen', '31 12 3555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (146, 'Saveley & Henriot, Co.', 'Mary ', 'Saveley', '78.32.5555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (148, 'Dragon Souveniers, Ltd.', 'Eric', 'Natividad', '+65 221 7555', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (151, 'Muscle Machine Inc', 'Jeff', 'Young', '2125557413', 1286, 'Tseng,Foon
Yue', 'ftseng@classicmodelcars.com'),
 (157, 'Diecast Classics Inc.', 'Kelvin', 'Leong', '2155551555', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (161, 'Technics Stores Inc.', 'Juri', 'Hashimoto', '6505556809', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (166, 'Handji Gifts& Co', 'Wendy', 'Victorino', '+65 224 1555', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com'),
 (167, 'Herkku Gifts', 'Veysel', 'Oeztan', '+47 2267 3215', 1504, 'Jones,Barry',
'bjones@classicmodelcars.com'),
 (168, 'American Souvenirs Inc', 'Keith', 'Franco', '2035557845', 1286,
'Tseng,Foon Yue', 'ftseng@classicmodelcars.com'),
 (171, 'Daedalus Designs Imports', 'Martine ', 'Rancé', '20.16.1555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (172, "La Corne D'abondance, Co.", 'Marie', 'Bertrand', '(1) 42.34.2555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (173, 'Cambridge Collectables Co.', 'Jerry', 'Tseng', '6175555555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (175, 'Gift Depot Inc.', 'Julie', 'King', '2035552570', 1323, 'Vanauf,George',
'gvanauf@classicmodelcars.com'),
 (177, 'Osaka Souveniers Co.', 'Mory', 'Kentary', '+81 06 6342 5555', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (181, 'Vitachrome Inc.', 'Michael', 'Frick', '2125551500', 1286, 'Tseng,Foon
Yue', 'ftseng@classicmodelcars.com'),
 (186, 'Toys of Finland, Co.', 'Matti', 'Karttunen', '90-224 8555', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (187, 'AV Stores, Co.', 'Rachel', 'Ashworth', '(171) 555-1555', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (189, 'Clover Collections, Co.', 'Dean', 'Cassidy', '+353 1862 1555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (198, 'Auto-Moto Classics Inc.', 'Leslie', 'Taylor', '6175558428', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (201, 'UK Collectables, Ltd.', 'Elizabeth', 'Devon', '(171) 555-2282', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (202, 'Canadian Gift Exchange Network', 'Yoshi ', 'Tamuri', '(604) 555-3392',
1323, 'Vanauf,George', 'gvanauf@classicmodelcars.com'),
 (204, 'Online Mini Collectables', 'Miguel', 'Barajas', '6175557555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (205, 'Toys4GrownUps.com', 'Julie', 'Young', '6265557265', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (209, 'Mini Caravy', 'Frédérique ', 'Citeaux', '88.60.1555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (211, 'King Kong Collectables, Co.', 'Mike', 'Gao', '+852 2251 1555', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (216, 'Enaco Distributors', 'Eduardo ', 'Saavedra', '(93) 203 4555', 1702,
'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (219, 'Boards & Toys Co.', 'Mary', 'Young', '3105552373', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (227, 'Heintze Collectables', 'Palle', 'Ibsen', '86 21 3555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (233, 'Québec Home Shopping Network', 'Jean ', 'Fresnière', '(514) 555-8054',
1286, 'Tseng,Foon Yue', 'ftseng@classicmodelcars.com'),
 (239, 'Collectable Mini Designs Co.', 'Valarie', 'Thompson', '7605558146',
1166, 'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (240, 'giftsbymail.co.uk', 'Helen ', 'Bennett', '(198) 555-8888', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (242, 'Alpha Cognac', 'Annette ', 'Roulet', '61.77.6555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (249, 'Amica Models & Co.', 'Paolo ', 'Accorti', '011-4988555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (250, 'Lyon Souveniers', 'Daniel', 'Da Silva', '+33 1 46 62 7555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (256, 'Auto Associés & Cie.', 'Daniel ', 'Tonini', '30.59.8555', 1370,
'Hernandez,Gerard', 'ghernande@classicmodelcars.com'),
 (259, 'Toms Spezialitäten, Ltd', 'Henriette ', 'Pfalzheim', '0221-5554327',
1504, 'Jones,Barry', 'bjones@classicmodelcars.com'),
 (260, 'Royal Canadian Collectables, Ltd.', 'Elizabeth ', 'Lincoln', '(604)
555-4555', 1323, 'Vanauf,George', 'gvanauf@classicmodelcars.com'),
 (276, "Anna's Decorations, Ltd", 'Anna', "O'Hara", '02 9936 8555', 1611,
'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (278, 'Rovelli Gifts', 'Giovanni ', 'Rovelli', '035-640555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (282, 'Souveniers And Things Co.', 'Adrian', 'Huxley', '+61 2 9495 8555', 1611,
'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (286, "Marta's Replicas Co.", 'Marta', 'Hernandez', '6175558555', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (298, 'Vida Sport, Ltd', 'Mihael', 'Holz', '0897-034555', 1702,
'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (299, 'Norway Gifts By Mail, Co.', 'Jan', 'Klaeboe', '+47 2212 1555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (311, 'Oulu Toy Supplies, Inc.', 'Pirkko', 'Koskitalo', '981-443655', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (314, 'Petit Auto', 'Catherine ', 'Dewey', '(02) 5554 67', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (319, 'Mini Classics', 'Steve', 'Frick', '9145554562', 1323, 'Vanauf,George',
'gvanauf@classicmodelcars.com'),
 (320, 'Mini Creations Ltd.', 'Wing', 'Huang', '5085559555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (321, 'Corporate Gift Ideas Co.', 'Julie', 'Brown', '6505551386', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (323, 'Down Under Souveniers, Inc', 'Mike', 'Graham', '+64 9 312 5555', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com'),
 (324, 'Stylish Desk Decors, Co.', 'Ann ', 'Brown', '(171) 555-0297', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (328, 'Tekni Collectables Inc.', 'William', 'Brown', '2015559350', 1323,
'Vanauf,George', 'gvanauf@classicmodelcars.com'),
 (333, 'Australian Gift Network, Co', 'Ben', 'Calaghan', '61-7-3844-6555', 1611,
'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (334, 'Suominen Souveniers', 'Kalle', 'Suominen', '+358 9 8045 555', 1501,
'Bott,Larry', 'lbott@classicmodelcars.com'),
 (339, 'Classic Gift Ideas, Inc', 'Francisca', 'Cervantes', '2155554695', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (344, 'CAF Imports', 'Jesus', 'Fernandez', '+34 913 728 555', 1702,
'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (347, "Men 'R' US Retailers, Ltd.", 'Brian', 'Chandler', '2155554369', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (350, 'Marseille Mini Autos', 'Laurence ', 'Lebihan', '91.24.4555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (353, 'Reims Collectables', 'Paul ', 'Henriot', '26.47.1555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (357, 'GiftsForHim.com', 'Wales', 'MacKinlay', '64-9-3763555', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com'),
 (362, 'Gifts4AllAges.com', 'Juri', 'Yoshido', '6175559555', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (363, 'Online Diecast Creations Co.', 'Dorothy', 'Young', '6035558647', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (376, 'Precious Collectables', 'Braun', 'Urs', '0452-076555', 1702,
'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (379, 'Collectables For Less Inc.', 'Allen', 'Nelson', '6175558555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (381, 'Royale Belge', 'Pascale ', 'Cartrain', '(071) 23 67 2555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (382, 'Salzburg Collectables', 'Georg ', 'Pipps', '6562-9555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (385, 'Cruz & Sons Co.', 'Arnold', 'Cruz', '+63 2 555 3587', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (386, "L'ordine Souveniers", 'Maurizio ', 'Moroni', '0522-556555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (398, 'Tokyo Collectables, Ltd', 'Akiko', 'Shimamura', '+81 3 3584 0555', 1621,
'Nishi,Mami', 'mnishi@classicmodelcars.com'),
 (406, 'Auto Canal+ Petit', 'Dominique', 'Perrier', '(1) 47.55.6555', 1337,
'Bondur,Loui', 'lbondur@classicmodelcars.com'),
 (412, 'Extreme Desk Decorations, Ltd', 'Sarah', 'McRoy', '04 499 9555', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com'),
 (415, 'Bavarian Collectables Imports, Co.', 'Michael', 'Donnermeyer', ' +49 89
61 08 9555', 1504, 'Jones,Barry', 'bjones@classicmodelcars.com'),
 (424, 'Classic Legends Inc.', 'Maria', 'Hernandez', '2125558493', 1286,
'Tseng,Foon Yue', 'ftseng@classicmodelcars.com'),
 (447, 'Gift Ideas Corp.', 'Dan', 'Lewis', '2035554407', 1323, 'Vanauf,George',
'gvanauf@classicmodelcars.com'),
 (448, 'Scandinavian Gift Ideas', 'Martha', 'Larsson', '0695-34 6555', 1504,
'Jones,Barry', 'bjones@classicmodelcars.com'),
 (450, 'The Sharp Gifts Warehouse', 'Sue', 'Frick', '4085553659', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (452, 'Mini Auto Werke', 'Roland ', 'Mendel', '7675-3555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (455, 'Super Scale Inc.', 'Leslie', 'Murphy', '2035559545', 1286, 'Tseng,Foon
Yue', 'ftseng@classicmodelcars.com'),
 (456, 'Microscale Inc.', 'Yu', 'Choi', '2125551957', 1286, 'Tseng,Foon Yue',
'ftseng@classicmodelcars.com'),
 (458, 'Corrida Auto Replicas, Ltd', 'Martín ', 'Sommer', '(91) 555 22 82',
1702, 'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (462, 'FunGiftIdeas.com', 'Violeta', 'Benitez', '5085552555', 1216,
'Patterson,Steve', 'spatterson@classicmodelcars.com'),
 (471, 'Australian Collectables, Ltd', 'Sean', 'Clenahan', '61-9-3844-6555',
1611, 'Fixter,Andy', 'afixter@classicmodelcars.com'),
 (473, 'Frau da Collezione', 'Franco', 'Ricotti', '+39 022515555', 1401,
'Castillo,Pamela', 'pcastillo@classicmodelcars.com'),
 (475, 'West Coast Collectables Co.', 'Steve', 'Thompson', '3105553722', 1166,
'Thompson,Leslie', 'lthompson@classicmodelcars.com'),
 (484, 'Iberia Gift Imports, Corp.', 'José Pedro ', 'Roel', '(95) 555 82 82',
1702, 'Gerard,Martin', 'mgerard@classicmodelcars.com'),
 (486, 'Motor Mint Distributors Inc.', 'Rosa', 'Salazar', '2155559857', 1323,
'Vanauf,George', 'gvanauf@classicmodelcars.com'),
 (487, 'Signal Collectibles Ltd.', 'Sue', 'Taylor', '4155554312', 1165,
'Jennings,Leslie', 'ljennings@classicmodelcars.com'),
 (489, 'Double Decker Gift Stores, Ltd', 'Thomas ', 'Smith', '(171) 555-7555',
1501, 'Bott,Larry', 'lbott@classicmodelcars.com'),
 (495, 'Diecast Collectables', 'Valarie', 'Franco', '6175552555', 1188,
'Firrelli,Julie', 'jfirrelli@classicmodelcars.com'),
 (496, "Kelly's Gift Shop", 'Tony', 'Snowden', '+64 9 5555500', 1612,
'Marsh,Peter', 'pmarsh@classicmodelcars.com')]
```

Note to self about view expansion: View expansion is the process by which the database system (or you, manually) replaces a reference to a view with the actual SQL query that defines that view. This is done recursively, meaning if a view uses another view, the system keeps expanding until only base tables remain.

In this question, we'd technically need to hw3_customersAndSales and go on to expand hw3_customerSummary to get the final query.

3.5 View Updates

Question

Consider the following SQL DDL based on a modification of the `db_book` data model.

```
drop table if exists hw3_student;

create table if not exists db_book.hw3_student
(
    ID        varchar(5)  not null
        primary key,
    name      varchar(20) not null,
    dept_name varchar(20) null,
    tot_cred  decimal(3)  not null
);

create or replace view db_book.hw3_student_simple_1 as
    select ID, name, tot_cred from hw3_student;
create or replace view db_book.hw3_student_simple_2 as
    select ID, name from hw3_student;
```

For each views, state whether or not it is updateable and why.

Answer

In general, a view is updatable if: • It is derived from a single base table. • It does not include: Aggregation (SUM, COUNT, etc.), DISTINCT, GROUP BY, HAVING, or UNION, Subqueries in the SELECT list, Set operations • It includes all columns necessary to uniquely identify a row in the base table — typically this means it includes the primary key.

hw3_student_simple_1 is updatable because it derives from a single table, does not use aggregation, and includes the primary key. INSERT, UPDATE, and DELETE operations can be performed through the view (subject to permission and schema constraints, of course).

hw3_student_simple_2 is also updatable, but it has some limitations since it doesn't include tot_cred, which is a required field. Unless the supports triggers or INSTEAD OF rules to handle missing columns, you can't insert new rows through this view.

Note: When inserting in a view that does not include a certain column, it will just insert NULL, but this fails in simple_2 because tot_creds is NOT NULL.

3.6 Fun with Joins

Question

Using the default dataset UIBK - R, S, T, execute the following relational algebra expression (a full outer join):

```
S   T
```

The result is:

Write and execute a relational algebra expression that produces an equivalent result but does not use outer join.

Answer

Replace the statement and image below with your results.

```
S   T
```

3.7 Indexes

Question

Properly identifying and defining useful indexes is critical to database performance. Briefly explain: 1. Why do some databases automatically create indexes when the DDL defines primary, unique and foreign key constraints? 2. If indexes are useful, why doesn't the database engine automatically create every possible index? 3. The following is the DDL for `student` from the `db_book` database. For the purposes of this question, a "scenario" is a hypothesis about the size of tables and the types of queries users submit. Define a scenario for which creating an additional index on `student` would be beneficial and why.

```
create table if not exists db_book.student
(
    ID        varchar(5)  not null
        primary key,
    name      varchar(20) not null,
    dept_name varchar(20) null,
    tot_cred  decimal(3)  null,
    constraint student_ibfk_1
        foreign key (dept_name) references db_book.department (dept_name)
            on delete set null,
    check (`tot_cred` >= 0)
);

create index dept_name
    on db_book.student (dept_name);
```

Answer

1. Databases automatically create indexes for primary, unique, and foreign keys because these constraints require fast lookups and enforcement of uniqueness or referential integrity, which would be inefficient without an index.
2. Even though indices may be useful, they come with costs — both in space and performance overhead during write operations. Every index takes disk space. Every data change must update all relevant indexes, slowing write performance. Too many indexes can confuse the optimizer and degrade performance.
3. A scenario: Suppose the university database serves an academic dashboard that frequently runs queries to track academic progress, like: SELECT * FROM student WHERE tot_cred >= 120; or SELECT AVG(tot_cred) FROM student WHERE dept_name = 'Physics';, then we'd want an index on tot_cred because these queries filter on or group by tot_cred, especially if looking for students who are near graduation thresholds (e.g., 120 credits)

3.8 Codd's Rule 7

Question

[ ]:

```
```

Codd's **Rule 7: High-Level Insert, Update, and Delete Rule**

A database must support high-level insertion, updation, and deletion. This must not be limited to a single row, … …

Briefly explain and give examples of how SQL realizes/implements this rule.

Answer

This rule emphasizes that a relational database system should allow set-oriented (not just row-at-a-time) data manipulation using high-level, declarative operations. SQL is a declarative, set-based language, and it fully supports this rule. It allows operations that affect multiple rows at once, rather than requiring procedural iteration over individual tuples. Additionally, we can do multi-value insert with multiple rows, you can update multiple rows at once (with 'SET Val = Val +3)', etc.

3.9 Complex Attributes

Question

Consider the following Python code that processes some information about *Game of Thrones.*

[19]:

```
fn = "s2025_episodes.json"

with open(fn, "r") as in_file:
    got_episodes = json.load(in_file)

for g in got_episodes:
    g["openingSequenceLocations"] = ";".join(g["openingSequenceLocations"])

got_episodes_df = pandas.DataFrame(got_episodes)
```

```
got_episodes_df
```

```
[19]:     seasonNum  episodeNum                           episodeTitle  \
      0           1           1                       Winter Is Coming
      1           1           2                         The Kingsroad
      2           1           3                             Lord Snow
      3           1           4  Cripples, Bastards, and Broken Things
      4           1           5                  The Wolf and the Lion
      ..        …         …                                      …
      68          8           2         A Knight of the Seven Kingdoms
      69          8           3                        The Long Night
      70          8           4                   The Last of the Starks
      71          8           5                              The Bells
      72          8           6                        The Iron Throne

         episodeAirDate                        openingSequenceLocations
      0      2011-04-17          King's Landing;Winterfell;The Wall;Pentos
      1      2011-04-24    King's Landing;Winterfell;The Wall;Vaes Dothrak
      2      2011-05-01    King's Landing;Winterfell;The Wall;Vaes Dothrak
      3      2011-05-08    King's Landing;Winterfell;The Wall;Vaes Dothrak
      4      2011-05-15  King's Landing;The Eyrie;Winterfell;The Wall;V…
      ..            …                                               …
      68     2019-04-21            Last Hearth;Winterfell;King's Landing
      69     2019-04-28            Last Hearth;Winterfell;King's Landing
      70     2019-05-05            Last Hearth;Winterfell;King's Landing
      71     2019-05-12            Last Hearth;Winterfell;King's Landing
      72     2019-05-19            Last Hearth;Winterfell;King's Landing

      [73 rows x 5 columns]
```

Assume that I want to store the resulting dataframe in a relational database.

Draw a Crow's Foot logical ER diagram for the data model I should use. Replace the image below with your image. You may add explanatory notes if you wish.

Answer

GoT Diagram

3.10 Two Critical Concepts in Relationships

Question

Two critical concepts in entity relationships are *degree* and *cardinality*. Briefly explain the concepts.

Answer

1. The degree of a relationship refers to the number of entity types that participate in the relationship. • Binary (degree = 2): most common; involves two entities. Example: Student enrolls in Course • Ternary (degree = 3): involves three entities. Example: Doctor prescribes Medication to Patient
2. Cardinality defines the number of instances of one entity that can be associated with instances of another entity. • One-to-One (1:1) Example: Each Person has one Passport, and each Passport belongs to one Person. • One-to-Many (1:N) Example: One Department has many Employees.

4 Practical Questions

4.1 Metadata

Question

The following MySQL operation generates a basic description of a table.

[20]:
```
%sql describe db_book.student;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
4 rows affected.
```

[20]:
```
[('ID', 'varchar(5)', 'NO', 'PRI', None, ''),
 ('name', 'varchar(20)', 'NO', '', None, ''),
 ('dept_name', 'varchar(20)', 'YES', 'MUL', None, ''),
 ('tot_cred', 'decimal(3,0)', 'YES', '', None, '')]
```

Examine the `views` and `tables` in `information_schema`. Write a simple SQL DML query that produces the information above. Your answer does not have to have the same format.

Answer SELECT COLUMN_NAME AS Field, COLUMN_TYPE AS Type, IS_NULLABLE AS Null, COLUMN_KEY AS `Key`, COLUMN_DEFAULT AS Default, EXTRA AS Extra FROM information_schema.COLUMNS WHERE TABLE_SCHEMA = 'db_book' AND TABLE_NAME = 'student';

Please place and execute your SQL statement below.

[19]:
```
%%sql
SELECT
    COLUMN_NAME AS Field,
    COLUMN_TYPE AS Type,
    IS_NULLABLE AS `Null`,
    COLUMN_KEY AS `Key`,
    COLUMN_DEFAULT AS `Default`,
    EXTRA AS Extra
FROM
    information_schema.COLUMNS
WHERE
    TABLE_SCHEMA = 'db_book'
    AND TABLE_NAME = 'student';
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
4 rows affected.
```

[19]:

```
[('ID', 'varchar(5)', 'NO', 'PRI', None, ''),
 ('name', 'varchar(20)', 'NO', '', None, ''),
 ('dept_name', 'varchar(20)', 'YES', 'MUL', None, ''),
 ('tot_cred', 'decimal(3,0)', 'YES', '', None, '')]
```

[15]:

```
%%sql
USE db_book;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
0 rows affected.
```

[15]:

```
[]
```

Note to self: Back-ticks are very important for Null, Default, and Key!

4.2 Tricky SQL 1

Question

Consider the following query. **Note** that I used `limit 10` just to keep the answer short.

[20]:

```
%%sql

use classicmodels;

with one as (
    select customerNumber, count(*) as numberOfOrders,
           sum(quantityOrdered*priceEach) as totalRevenue
    from orders join orderdetails using(orderNumber) group by customerNumber
)
select * from one
limit 10;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
0 rows affected.
10 rows affected.
```

[20]:

```
[(103, 7, Decimal('22314.36')),
 (112, 29, Decimal('80180.98')),
 (114, 55, Decimal('180585.07')),
 (119, 53, Decimal('158573.12')),
 (121, 32, Decimal('104224.79')),
 (124, 180, Decimal('591827.34')),
 (128, 22, Decimal('75937.76')),
 (129, 21, Decimal('66710.56')),
 (131, 49, Decimal('149085.15')),
 (141, 259, Decimal('820689.54'))]
```

Modify the query so that the results contains: - The customer's `customerName` and the customer's `country`. - Only contains customers whose country is `Spain`

You must not use a JOIN.

Answer

[21]:

```
%%sql

USE classicmodels;

WITH one AS (
    SELECT
        customerNumber,
        COUNT(*) AS numberOfOrders,
        SUM(quantityOrdered * priceEach) AS totalRevenue
    FROM orders
    JOIN orderdetails USING (orderNumber)
    GROUP BY customerNumber
)
SELECT
    customerNumber,
    (SELECT customerName FROM customers WHERE customers.customerNumber = one.
  ↪customerNumber) AS customerName,
    (SELECT country FROM customers WHERE customers.customerNumber = one.
  ↪customerNumber) AS country,
    numberOfOrders,
    totalRevenue
FROM one
WHERE (SELECT country FROM customers WHERE customers.customerNumber = one.
  ↪customerNumber) = 'Spain';
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
0 rows affected.
5 rows affected.
```

[21]:

```
[(141, 'Euro+ Shopping Channel', 'Spain', 259, Decimal('820689.54')),
 (216, 'Enaco Distributors', 'Spain', 23, Decimal('68520.47')),
 (344, 'CAF Imports', 'Spain', 13, Decimal('46751.14')),
 (458, 'Corrida Auto Replicas, Ltd', 'Spain', 32, Decimal('112440.09')),
 (484, 'Iberia Gift Imports, Corp.', 'Spain', 15, Decimal('50987.85'))]
```

4.3 Tricky SQL 2

Question

Use the `db_book` database for this problem.

We want to compute a relation of the form $(course\_id, course\_title, prereq\_course\_id, prereq\_course\_title)$ that lists courses and their prereqs. If a course does not have a prereq, $prereq\_course\_id$ and *prereq_course_title* should be `NULL`. If a course is not a prereq for any other course, it should appear in the result with *course_id* and *course_title* both `NULL`.

Write an SQL query that produces the table. To help, the result of running my solution is below.

Course-Prereqs

Answer

```
[24]: %%sql
SELECT
    c.course_id,
    c.title AS course_title,
    p.prereq_id AS prereq_course_id,
    (SELECT title FROM course WHERE course.course_id = p.prereq_id) AS␣
 ↪prereq_course_title
FROM
    course c
LEFT JOIN
    prereq p ON c.course_id = p.course_id

UNION

SELECT
    NULL AS course_id,
    NULL AS course_title,
    p.prereq_id AS prereq_course_id,
    c.title AS prereq_course_title
FROM
    prereq p
LEFT JOIN
    course c ON p.prereq_id = c.course_id
WHERE
    p.prereq_id NOT IN (SELECT course_id FROM prereq);
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
16 rows affected.
```

```
[24]: [('BIO-101', 'Intro. to Biology', None, None),
 ('BIO-301', 'Genetics', 'BIO-101', 'Intro. to Biology'),
 ('BIO-399', 'Computational Biology', 'BIO-101', 'Intro. to Biology'),
 ('CS-101', 'Intro. to Computer Science', None, None),
 ('CS-190', 'Game Design', 'CS-101', 'Intro. to Computer Science'),
 ('CS-315', 'Robotics', 'CS-101', 'Intro. to Computer Science'),
 ('CS-319', 'Image Processing', 'CS-101', 'Intro. to Computer Science'),
 ('CS-347', 'Database System Concepts', 'CS-101', 'Intro. to Computer Science'),
 ('EE-181', 'Intro. to Digital Systems', 'PHY-101', 'Physical Principles'),
 ('FIN-201', 'Investment Banking', None, None),
 ('HIS-351', 'World History', None, None),
 ('MU-199', 'Music Video Production', None, None),
 ('PHY-101', 'Physical Principles', None, None),
 (None, None, 'BIO-101', 'Intro. to Biology'),
 (None, None, 'CS-101', 'Intro. to Computer Science'),
 (None, None, 'PHY-101', 'Physical Principles')]
```

[23]:
```
%%sql
USE db_book;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
0 rows affected.
```

[23]:
```
[]
```

4.4 ER Diagram and DDL

Question

Consider the following scenario. There are 6 entity types. 1. BaseProduct that has attributes: - product_id - product_name - product_description - price 2. DurableProduct **isA** BaseProduct and has the attributes: - height - width - depth - weight 3. EntertainmentProduct **isA** a BaseProduct and has the attributes: - copyright_date - version_no - artist_first_name - artist_last_name 4. MusicProduct **isA** an EntertainmentProduct an has the attributes: - no_of_seconds - genre 5. BookProduct **isA** an EntertainmentProduct and has the attributes: - no_of_pages - printing_no 6. Supplier has the attributes: - supplier_id - supplier_name

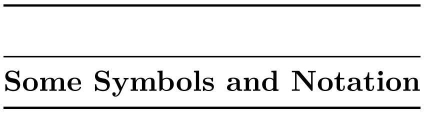
A product is either a DurableProduct, an EntertainmentProduct or just a BaseProduct. There are no products that are both durable and entertainment. All EntertainmentProducts are either a book or a music product but not both. Every product has *exactly* one Supplier. A Supplier may supply several products.

Draw a Crow's Foot ER Diagram for the logical model for this scenario. You may have to add attributes to the lists above to implement define the model. Specify whether you are choosing the one, two or three table design. You should also specify which views you would create.

You should indicate **all foreign keys.**

Answer

Place a screenshot of you diagram here and replace this image, which is some symbols.



Some Symbols and Notation

Please place and execute your SQL statement below. CREATE VIEW FullProduct AS SELECT bp.product_id, bp.product_name, bp.price, s.supplier_name, dp.height, dp.width, dp.depth, dp.weight, ep.copyright_date, ep.artist_first_name, mp.genre, bp.product_description FROM baseproduct bp JOIN supplier s ON bp.supplier_id = s.supplier_id LEFT JOIN durableproduct dp ON bp.product_id = dp.product_id LEFT JOIN entertainmentproduct ep ON bp.product_id = ep.product_id LEFT JOIN musicproduct mp ON ep.product_id = mp.product_id LEFT JOIN bookproduct bk ON ep.product_id = bk.product_id;

I went for a view that combines all the tables into one singular table using join and left, to preserve the base information, adding null for any additional columns that relate to the other specializations.

[27]:
```
%%sql
CREATE VIEW FullProduct AS
SELECT
    bp.product_id,
    bp.product_name,
    bp.price,
    s.supplier_name,
    dp.height, dp.width, dp.depth, dp.weight,
    ep.copyright_date, ep.artist_first_name,
    mp.genre, bp.product_description
FROM baseproduct bp
JOIN supplier s ON bp.supplier_id = s.supplier_id
LEFT JOIN durableproduct dp ON bp.product_id = dp.product_id
LEFT JOIN entertainmentproduct ep ON bp.product_id = ep.product_id
LEFT JOIN musicproduct mp ON ep.product_id = mp.product_id
LEFT JOIN bookproduct bk ON ep.product_id = bk.product_id;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
(pymysql.err.ProgrammingError) (1146, "Table 'classicmodels.baseproduct' doesn't
exist")
[SQL: CREATE VIEW FullProduct AS
SELECT
    bp.product_id,
    bp.product_name,
    bp.price,
    s.supplier_name,
    dp.height, dp.width, dp.depth, dp.weight,
    ep.copyright_date, ep.artist_first_name,
    mp.genre, bp.product_description
FROM baseproduct bp
JOIN supplier s ON bp.supplier_id = s.supplier_id
LEFT JOIN durableproduct dp ON bp.product_id = dp.product_id
LEFT JOIN entertainmentproduct ep ON bp.product_id = ep.product_id
LEFT JOIN musicproduct mp ON ep.product_id = mp.product_id
LEFT JOIN bookproduct bk ON ep.product_id = bk.product_id;]
(Background on this error at: https://sqlalche.me/e/20/f405)
```

4.5 I Told You that You Would Have to Look Up Functions SQL Question

Question

The following result table uses `section` and `time_slot` from `db_book`.

Using Functions

Write an SQL query that produces the table. You will have to look for some SQL functions to use. You can use the web, ChatGPT, etc. to find the documentation on the functions.

The order of the multiple day classes must be “MWF” or “TR.” Other orders, e.g. “FMW” are not acceptable.

[28]:

```
%%sql
USE db_book;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
0 rows affected.
```

[28]: []

Answer

[30]:

```
%%sql
CREATE TABLE IF NOT EXISTS db_book.day_patterns (
    pattern_name VARCHAR(10) PRIMARY KEY,
day_list VARCHAR(10) NOT NULL
);
INSERT INTO db_book.day_patterns (pattern_name, day_list)
VALUES
    ('MWF', 'M,W,F'),
    ('TR', 'T,R'),
    ('MW', 'M,W'),
    ('WF', 'W,F'),
    ('W', 'W'),
    ('F', 'F'),
    ('M', 'M'),
    ('T', 'T'),
    ('R', 'R');
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
0 rows affected.
9 rows affected.
```

[30]: []

[33]:

```
%%sql
WITH grouped_times AS (
    SELECT
        s.course_id,
        s.sec_id,
        s.semester,
        s.year,
        ts.time_slot_id,
        -- Format start_time and end_time as HH:MM
        TIME_FORMAT(MAKETIME(ts.start_hr, ts.start_min, 0), '%H:%i') AS␣
 ↪start_time,
        TIME_FORMAT(MAKETIME(ts.end_hr, ts.end_min, 0), '%H:%i') AS end_time,
        GROUP_CONCAT(ts.day ORDER BY FIELD(ts.day, 'M', 'T', 'W', 'R', 'F')␣
 ↪SEPARATOR ',') AS raw_day_string
    FROM db_book.section s
    JOIN db_book.time_slot ts ON s.time_slot_id = ts.time_slot_id
    GROUP BY
        s.course_id, s.sec_id, s.semester, s.year,
        ts.time_slot_id, ts.start_hr, ts.start_min, ts.end_hr, ts.end_min
)
SELECT
    g.course_id,
    g.sec_id,
    g.semester,
    g.year,
    dp.pattern_name AS days,
    g.start_time,
    g.end_time
FROM grouped_times g
LEFT JOIN db_book.day_patterns dp ON g.raw_day_string = dp.day_list
ORDER BY g.year, g.semester, g.course_id;
```

```
 * mysql+pymysql://root:***@localhost?local_infile=1
15 rows affected.
```

[33]:

```
[('CS-101', '1', 'Fall', Decimal('2017'), 'W', '10:00', '12:30'),
 ('CS-347', '1', 'Fall', Decimal('2017'), 'MWF', '08:00', '08:50'),
 ('PHY-101', '1', 'Fall', Decimal('2017'), 'MWF', '08:00', '08:50'),
 ('CS-190', '1', 'Spring', Decimal('2017'), 'TR', '10:30', '11:45'),
 ('CS-190', '2', 'Spring', Decimal('2017'), 'MWF', '08:00', '08:50'),
 ('EE-181', '1', 'Spring', Decimal('2017'), 'MWF', '11:00', '11:50'),
 ('BIO-101', '1', 'Summer', Decimal('2017'), 'MWF', '09:00', '09:50'),
 ('CS-101', '1', 'Spring', Decimal('2018'), 'TR', '14:30', '15:45'),
 ('CS-315', '1', 'Spring', Decimal('2018'), 'MWF', '13:00', '13:50'),
 ('CS-319', '1', 'Spring', Decimal('2018'), 'MWF', '09:00', '09:50'),
 ('CS-319', '2', 'Spring', Decimal('2018'), 'MWF', '11:00', '11:50'),
 ('FIN-201', '1', 'Spring', Decimal('2018'), 'MWF', '09:00', '09:50'),
 ('HIS-351', '1', 'Spring', Decimal('2018'), 'MWF', '11:00', '11:50'),
 ('MU-199', '1', 'Spring', Decimal('2018'), 'MWF', '13:00', '13:50'),
 ('BIO-301', '1', 'Summer', Decimal('2018'), 'MWF', '08:00', '08:50')]
```

[ ]: