**Practice Exercises**

For practice questions 1 to 6, consider the following dataset:

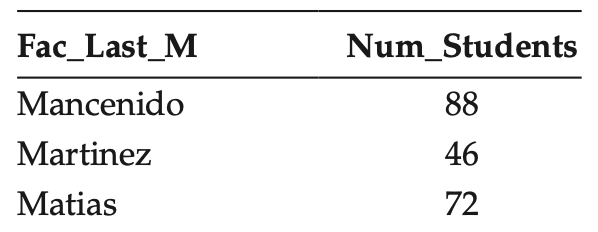
A screenshot of a cell phone

Description automatically generated

1. Create a query to count the number of students that took Supply Chain

Management. Name the column “Student\_Count.”

1. Create a query to handle the following desired query result:



3. Create a query for the following report:

A screenshot of a cell phone

Description automatically generated

1. The report in question 3 seems to lack the actual description of the faculty rank. Create a table named “Rank\_Description” with the attributes:
   * *Fac\_Rank\_C: A four-character attribute with default value INST and cannot be NULL. Additionally, this is the primary key of the table.*
   * *Fac\_Rank\_M: A variable character attribute that describes the rank.  
     • Min\_Sal\_Grade: A numeric 1-digit number that can take values from 1 to 9,*

*indicating the minimum salary grade of the rank.*

1. Write an SQL command to populate the table in question 4 with the following data:

A screenshot of a cell phone

Description automatically generated

6. Write a nested query without join statements for the following report:

A screenshot of a cell phone

Description automatically generated

For questions 7 to 12, consider the following tables:

A screenshot of a cell phone

Description automatically generated

1. The *PO\_Pay\_Terms* will be used in different tables; hence, a domain could be use- ful to handle this in the event of future changes. This attribute must contain a maximum of 10 characters; the default value should be “NET30,” which corre- sponds to net 30 days; and should not be NULL. Write an SQL code to create a domain named “*Pay\_Terms*” subject to the aforementioned specifications.
2. Create a “Suppliers” table with the following attributes:
   * Supplier\_N: A four integer–digit primary key attribute used to uniquely iden- tify a supplier and should not be NULL.
   * *Supplier\_M: A string that denotes the name of a supplier with a maximum of 255 characters.*
   * *Supplier\_Tier: A single-digit number between 1 and 5 indicating the supplier’s tier.*
   * Default\_Pay\_Terms: An attribute that follows the domain “Pay\_Terms” defined

in question 6.

1. Populate the table generated in question 8 with the following information:

A screenshot of a cell phone

Description automatically generated

10. Create a query to for this report:

A screenshot of a cell phone

Description automatically generated

11. A round of supplier reviews has lapsed, and the Beltway Coffee supplier’s master data need to be updated (see bold font).

A screenshot of a cell phone

Description automatically generated

Create a single query to update the record.

12. A new attribute is needed for the *Purchase\_Orders* relation to denote whether the purchase order instance is paid. Write the corresponding SQL code to add the fol- lowing attributes:

• Pay\_Status: A three-character string indicating the status of the purchase order payment, with the default value “NYP” indicating not yet paid.

SQL Code:

Create database Project;

use Project;

CREATE TABLE faculty (

fac\_N int primary key,

fac\_First\_M VARCHAR(20),

fac\_Last\_M VARCHAR(20),

fac\_Rank\_C VARCHAR(7)

);

insert into faculty

values

(13443, 'Iris', 'Martinez', 'ASCP'),

(66234, 'Aura', 'Matias', 'PROF'),

(45463, 'Lorelie', 'Grepo', 'ASTP'),

(44556, 'Mickey','Mancenido','ASTP');

Create table courses (

course\_C varchar(20) primary key,

course\_M varchar(50),

course\_units int

);

insert into courses

values

('IDOE', 'Design of Experients', 3),

('IBDM', 'Business Intelligence and Data Mining', 3),

('MSCM', 'Supply Chain Management', 3),

('SRES', 'Research Methods', 1),

('SSIM', 'Simulation', 4);

create table faculty\_courses (

Course\_C varchar(20) references courses(course\_C),

Fac\_N int references faculty(fac\_N),

Student\_Count int,

Sem\_C varchar(20)

);

insert into faculty\_courses

values

('MSCM', 13443, 34, '2012-1'),

('MSCM', 66234, 32, '2012-2'),

('IDOE', 44556, 56, '2012-1'),

('SRES', 13443, 12, '2012-1'),

('IBDM', 66234, 40, '2012-2'),

('MSCM', 44556, 32, '2013-1');

SELECT \* FROM faculty;

select \* from courses;

select \* from faculty\_courses;

select Student\_Count from faculty\_courses where Course\_C = 'MSCM';

select faculty.fac\_N, fac\_Last\_M, sum(Student\_Count)

from faculty\_courses, faculty

where faculty\_courses.Fac\_N = faculty.fac\_N

group by fac\_Last\_M

order by fac\_Last\_M;

select course\_M, course\_units, Sem\_C, fac\_Last\_M, fac\_Rank\_C

from courses

inner join faculty\_courses

on courses.course\_C = faculty\_courses.Course\_C

inner join faculty

on faculty.fac\_N = faculty\_courses.Fac\_N;

Create table rank\_description (

fac\_rank\_C varchar(4) NOT NULL DEFAULT 'inst' primary key,

fac\_rank\_M varchar(50),

min\_sal\_grade int CHECK(min\_sal\_grade < 10)

);

insert into rank\_description

values

('INST','Instructor',1),

('ASTP','Assistant Professor',3),

('ASCP','Associate Professor',5),

('PROF','Professor',7),

('UPRF','University Professor',9);

select rank\_description.fac\_rank\_M, rank\_description.fac\_rank\_C,

from rank\_description

join faculty

on faculty.fac\_Rank\_C = rank\_description.fac\_rank\_C;

select rank\_description.fac\_rank\_M, rank\_description.fac\_rank\_C, faculty.fac\_First\_M

from rank\_description

where fac\_rank\_C in

(select fac\_Rank\_C

from faculty);

select fac\_First\_M, fac\_Last\_M

from faculty

where faculry.fac\_Rank\_C

(select fac\_rank\_C, Fac\_Rank\_M

from rank\_description);

select \* from rank\_description;

WHERE rank\_description.fac\_rank\_c = faculty.fac\_rank\_c

Round 2:

Create database Orders;

use Orders;

CREATE TABLE orders (

po\_n int primary key,

po\_d VARCHAR(20),

supplier\_n int,

po\_payterms VARCHAR(10) NOT NULL default 'NET30'

);

insert into orders

values

(610557, '2/27/2013', 1335, 'NET30'),

(610558, '2/27/2013', 2652, '2/10NET20'),

(610559, '2/27/2013', 1335, 'COD'),

(610560, '2/28/2013', 1226, '2/10NET20'),

(610561, '3/01/2013', 2652,'2/10NET20');

CREATE TABLE order\_items (

po\_n int references orders(po\_n),

item\_n int,

item\_q int,

item\_price decimal (7,2)

);

insert into order\_items

values

(610557, 36796, 15, 664.25),

(610557, 36224, 21, 224.54),

(610559, 36624, 100, 0.65),

(610560, 36547, 1, 10887.10),

(610561, 36869, 224, 336.65);

CREATE TABLE suppliers (

supplier\_n int primary key NOT NULL,

supplier\_m varchar(255),

supplier\_tier int CHECK(supplier\_tier < 6),

default\_payterms varchar(10) NOT NULL default 'NET30'

);

insert into suppliers (supplier\_n, supplier\_m, supplier\_tier)

values

(1335, 'MacLarens Irish Pub', 4),

(2652, 'Central Park', 3),

(1226, 'Beltway Coffee', Null);

select \* from suppliers;

select supplier\_m, orders.po\_n, item\_n, item\_q

from suppliers

join orders

on orders.supplier\_n = suppliers.supplier\_n

left join order\_items

on orders.po\_n = order\_items.po\_n;

update suppliers

set supplier\_tier = 3, default\_payterms = '2/10NET20'

where supplier\_n = 1226;

alter table orders

add pay\_status VARCHAR(3) default 'NYP';

select \* from orders;