**Answers**

**University of Pittsburgh**

**BUSBIS 1605**

Professor David Eargle

Mid Semester Examination

Spring 2017

By submitting this exam you acknowledge that the completed exam is entirely your own work and you have complied with the Academic Honesty program of the University of Pittsburgh.

Submitted exams are analyzed to identify similarities in content and structure.

You may consult your notes and textbook and use your computer for testing SQL queries or drawing data models. You can even search the web for general help with queries, syntax, etc. But you may not consult anyone else.

Please submit this exam on CourseWeb as a single PDF

# CourseWeb Quiz (2 points)

Before doing anything else, complete the short quiz on CourseWeb called “Midterm – short quiz.”

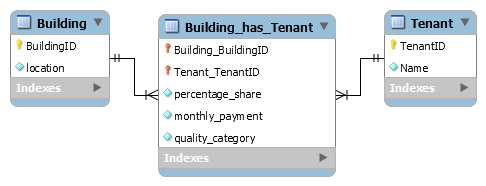
# Data Modeling

## Data modeling – part 1 (40 points)

Draw data models for the following two situations. Be thoughtful in the name you give each entity.

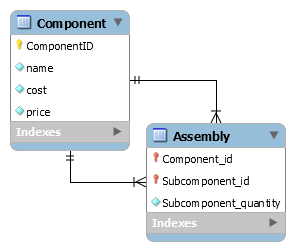
1. (20 points)

A building is leased by multiple tenants, and a tenant can lease multiple buildings. Each lease has a start date, percentage share of the building, and monthly payment amount. Additionally, each lease can be categorized as being premium-, standard-, or slum-quality.



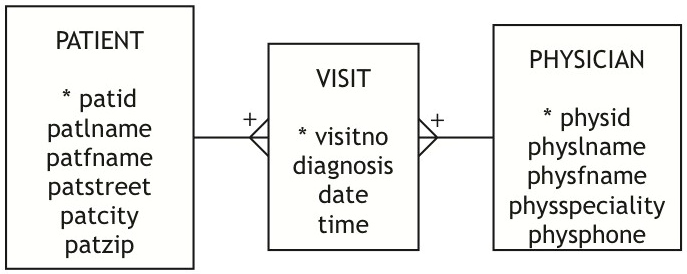
1. (20 points)

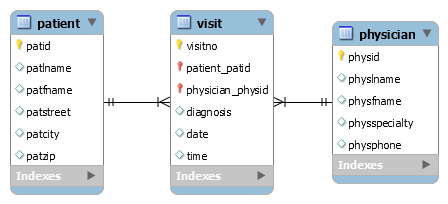
A computer part is made of many components, and a computer part can be a component of many other parts. Record the name of each part, as well as the quantity of the part used for each part-component combination.



## Data modeling – part 2 (12 points)

A physician’s office has developed the following data model for storing details of patient visits. When you map the data model to a relational database, what will be the columns, primary key, and foreign key(s) for the table Visit?





CREATE TABLE IF NOT EXISTS `visit` (

`visitno` INT NOT NULL,

`patient\_patid` INT NOT NULL,

`physician\_physid` INT NOT NULL,

`diagnosis` VARCHAR(45) NULL,

`date` VARCHAR(45) NULL,

`time` VARCHAR(45) NULL,

PRIMARY KEY (`visitno`, `patient\_patid`, `physician\_physid`),

INDEX `fk\_visit\_patient1\_idx` (`patient\_patid` ASC),

INDEX `fk\_visit\_physician1\_idx` (`physician\_physid` ASC),

CONSTRAINT `fk\_visit\_patient1`

FOREIGN KEY (`patient\_patid`)

REFERENCES `mydb`.`patient` (`patid`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_visit\_physician1`

FOREIGN KEY (`physician\_physid`)

REFERENCES `mydb`.`physician` (`physid`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

# SQL

Use the ClassicModels database to answer the following requests using SQL. The data model is on the [book web site](http://richardtwatson.com/dm6e/Reader/ClassicModels.html).

## SQL part 1 (16 points – 4 points each)

1. Select the number of offices from each country.

SELECT country, COUNT(\*)

FROM Offices

GROUP BY country;

1. Which Product has the highest MSRP?

SELECT productName FROM Products

WHERE MSRP = (SELECT MAX(MSRP) FROM Products);

1. Report details of those payments greater than the average of all payments.

SELECT \*

FROM payments

WHERE amount > (SELECT AVG(amount) FROM Payments);

1. What is the total value of payments received in February 2003 for each country?

SELECT country, SUM(amount)

FROM Payments

JOIN customers ON Payments.customerNumber = Customers.customerNumber

WHERE MONTH(paymentDate) = 2

AND YEAR(paymentDate) = 2003

GROUP BY country;

## SQL part 2 (30 points – 10 for each query)

1. How many distinct products were involved in a sale recorded in 2003? (Hint: No need to use DISTINCT for this.)

SELECT COUNT(distinct productCode) as distinct\_product

From OrderDetails

JOIN Orders

ON Orders.orderNumber = OrderDetails.orderNumber

WHERE YEAR(orderDate) = 2003

OR

SELECT COUNT(\*)

FROM Products

WHERE EXISTS (

SELECT \* FROM orderdetails JOIN Orders

ON orderDetails.orderNumber = Orders.orderNumber

WHERE orderdetails.productCode = Products.productCode

AND YEAR(orderDate) = 2003

);

OR

SELECT COUNT(\*) AS Products

FROM products

WHERE productCode IN (

SELECT productCode FROM orderdetails

JOIN orders ON

orders.orderNumber=orderdetails.orderNumber

WHERE YEAR(orderDate)=2003

)

1. Which products have a greater average quantity ordered than are currently in stock for that product?

SELECT

Products.productCode,

productName,

AVG(quantityOrdered),

Products.quantityInStock

FROM Products JOIN OrderDetails

ON Products.productCode = OrderDetails.productCode

GROUP BY Products.productCode

HAVING AVG(quantityOrdered) > Products.quantityInStock;

1. What are the maximum, average, and minimum **percentage discounts** off of MSRP that each product has ever been sold at? Format the percentage discounts to two decimal places. (Hint: For each product, calculate the ratio of sale price to MSRP for each of its line items, and subtract from 1 to get the percentage discount.)

SELECT

Products.productName,

FORMAT((1-MIN(priceEach/MSRP))\*100,2) AS MaxDiscount,

FORMAT((1-MAX(priceEach/MSRP))\*100,2) AS MinDiscount,

FORMAT((1-AVG(priceEach/MSRP))\*100,2) AS AvgDiscount

FROM Products JOIN orderDetails ON Products.productCode = orderDetails.ProductCode

GROUP BY OrderDetails.productCode;