

## CSC1023 Databases

### Practical 4 – Physical database design & SQL-1

#### Exercise A. Physical database design

##### Instruction

- Exercise A of this practical is about Physical database design and it can be done without additional software.
- Although you would need to have the **answers** to B3, C3 and D3 of **Practical 3** to help you with A3, A4 and A5 in this Practical 4.

**A1.** What is main objective of physical design for RDBMS?

**A2.** For the following fields (hopefully obvious what they are) in the table Customer build a data dictionary (by hand, or using the [template](#) that is available on CSC1023 CANVAS Modules page, under 'Practical – Resources' section).

Make sure you include the data type, length, if it's required/indexed/unique, and identify the keys.

- CustomerNo
- Firstname
- Surname
- BillingAddress
- Notes
- ShippingAddress
- CreditLimit
- AccountBalance

**A3.** Using the 3NF schema: Question **B3 of Practical 3**, create the necessary data dictionary form(s) for each table and field.

**A4.** Using the 3NF schema: Question **C3 of Practical 3**, create the necessary data dictionary form(s) for each table and field.

**A5.** Using the 3NF schema **D3 of Practical 3**, create the necessary data dictionary form(s) for each table and field.

## Exercises B to F. SQL-1

### Instruction

- Exercises B to F are about Structured Query Language (SQL), and they require you to make use of the EECS MySQL hosting account.

If you have not yet set up a MySQL hosting account, then please follow the instructions in the [“MySQL Account Installation”](#) to set up a EECS MySQL hosting account.

(Note: this is the same instruction available on CSC1023 CANVAS Modules page, under ‘Practical – Resources’ section with the file name of MySQL\_Accout\_Installation\_Guide.pdf.

Alternatively, you may choose to have MySQL hosted on your own PC (although we can only support the EECS hosted account).

- You will need to record your SQL statements and the output screens obtained on SQL-phpMyAdmin.
  - To capture the SQL statements used, simply copy and paste from SQL-phpMyAdmin.
  - To obtain screen captures of your MySQL-phpMyAdmin output screen (the Entity tables, rows or columns of attributes, etc), you can use the [“Snipping Tool](#) (for Windows OS)” or [‘control-shift-Command-4’](#) (for Mac OS)” to capture your output screen. To paste your screen captures into this document, use [‘Control-V’](#) (for Windows OS) or [‘Command-V’](#) (for Mac OS).

### Task:

- You must have created a EECS LAMP hosting account and be able to connect and [login](#) to [phpMyAdmin](#) on that (or your own installation).
- Login to phpMyAdmin and [select your database](#) (on the left pane).
- [Download](#) the [northwind-complete.sql](#) file to the PC that you are working on. If the link doesn’t work, go to CSC1023 CANVAS Modules page, under ‘Practical – Resources’ section, to download the same file.
- [Import](#) [northwind-complete.sql](#) file to your phpMyAdmin, by following this [video guide](#). If the link doesn’t work, go to CSC1023 CANVAS Modules page, under ‘Practical – Resources’ section, to watch the video.
- Alternatively, follow the instructions written below:
  - [Invoke](#)/open a [text editor software](#) of your choice (Notepad++ , Textedit, etc)
  - [Open](#) the [northwind-complete.sql](#) file through your text editor.

- Copy the entire file → Highlight everything in the file, apply (Control+A) and copy into the clipboard (Control+C)
- In [phpMyAdmin](#), open the phpMyAdmin's MySQL code editor, accessible via the 'SQL' tab along the top menu (on the pane), and paste the content of the clipboard (Control+V) onto the blank space (the input area of the SQL tab) on the screen.
- Click Go (bottom right)
- This should load the example tables in – click again on your database on the left to see if it has loaded them, you should see something like the following:

Table	Action	Rows	Type	Collation	Size	Overhead
customers	★ Browse Structure Search Insert Empty Drop	29	InnoDB	utf8_general_ci	112 KiB	-
employees	★ Browse Structure Search Insert Empty Drop	9	InnoDB	utf8_general_ci	112 KiB	-
employee_privileges	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	64 KiB	-
inventory_transactions	★ Browse Structure Search Insert Empty Drop	102	InnoDB	utf8_general_ci	128 KiB	-
inventory_transaction_types	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16 KiB	-
invoices	★ Browse Structure Search Insert Empty Drop	35	InnoDB	utf8_general_ci	64 KiB	-
orders	★ Browse Structure Search Insert Empty Drop	48	InnoDB	utf8_general_ci	192 KiB	-
orders_status	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16 KiB	-
orders_tax_status	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	16 KiB	-
order_details	★ Browse Structure Search Insert Empty Drop	58	InnoDB	utf8_general_ci	192 KiB	-
order_details_status	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_general_ci	16 KiB	-
privileges	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	16 KiB	-
products	★ Browse Structure Search Insert Empty Drop	45	InnoDB	utf8_general_ci	32 KiB	-
purchase_orders	★ Browse Structure Search Insert Empty Drop	28	InnoDB	utf8_general_ci	112 KiB	-
purchase_order_details	★ Browse Structure Search Insert Empty Drop	55	InnoDB	utf8_general_ci	128 KiB	-
purchase_order_status	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16 KiB	-
sales_reports	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_general_ci	16 KiB	-
shippers	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	112 KiB	-
strings	★ Browse Structure Search Insert Empty Drop	62	InnoDB	utf8_general_ci	16 KiB	-
suppliers	★ Browse Structure Search Insert Empty Drop	10	InnoDB	utf8_general_ci	112 KiB	-
20 tables	Sum	511	InnoDB	latin1_swedish_ci	1.5 MiB	0 B

Note that you can see the tables (20 here) created and also how many rows (records) are contained in each one, for example there are 29 customers and 9 employees.

## • Executing SQL Queries

Within phpMyAdmin when we want to perform an SQL query, the easiest option is to click on the **SQL tab** and **type** in the **query** then press **Go**.

Once you've performed a query and are seeing the results you can change the query as needed without going back by clicking "edit inline" and clicking Go to update.

Remember to paste your SQL and your result into this document.

## **Section B. Exploring the Northwind Database**

Perform the following tasks for all the questions listed in this section:

- **Be familiar with the with the structure (schema) and contents (data) of the Northwind database.**
- **Learn your way around phpMyAdmin**
- **Some answers can be found by clicking through the interface in phpMyAdmin**
- **Other answers may require the use of Google (or ask a demonstrator)**
- **Paste your answers into this document**

**B1.** How many tables are in the Northwind schema?

**B2.** How many rows are in the suppliers table?

**B3.** For the first supplier (id 1) what is the company name?

**B4.** What data type is the city column in the suppliers table?

**B5.** What is the table type of the orders\_status table?

**B6.** What is the size (storage size – KiB) of the order\_details table?

**B7.** How many columns are there in the employees table?

**B8.** What is the data type of the attachments column in the employees table?

**B9.** What's the definition of the data found in question 8?

**B10.** What column is the primary key of the employees table?

**B11.** What extra attributes does the id column of the products table have?

**B12.** What does the attribute you found in question 11 mean?

**B13.** What are the two tax status names stored in the `orders_tax_status` table?

**B14.** In the `orders` table how many different countries are contained within the `ship_country_region` column throughout all 48 rows?

## **Section C. SQL query using the `SELECT` statement**

Perform the following tasks for all the questions listed in this section:

- **write** an **SQL query**,
- **execute** the query,
- **Check** your output **table** gives the expected result,
- **paste** the **SQL query** statement into this document.

**C1.** List all the columns for all the rows in the `customers` table.

**C2.** List the `id` and `company` columns for all the rows in the `customers` table.

**C3.** List all the columns for all the rows in the `products` table.

**C4.** List the `product_code`, `product_name` and `standard_cost` for all the rows in the `products` table.

**C5.** List the `company` column for all the row(s) in the `customers` table with an `id` of 12.

**C6.** List the `product_code` and `product_name` for all products with a `standard_cost` greater than 10 in the `products` table.

**C7.** List the `product_code` for a product with the `product_name` of “Northwind Traders Scones” in the `products` table.

**C8.** List the `company`, `last_name` and `first_name` of all customers whose `job_title` starts with the word “purchasing” in the `customers` table.

**C9.** List the `company` and `business_phone` of all customers whose `job_title` contains the word “Manager” in the `customers` table.

**C10.** List the **company** of all customers who are based in California (the state\_province is CA) in the **customers** table.

**More challenging questions:**

**C11.** List the **product\_code** and **product\_name** for all very profitable products (very profitable is if the profit per item is more than 5 i.e. the list\_price minus the standard\_cost is more than five) in the **products** table.

**C12.** List the **product\_code** and **product\_name** for all products that have a standard\_cost between 10 and 20 inclusive in the **products** table.

See the next few pages for Exercises D to F.

## Section D. SQL Query using Built-in Aggregate Functions

Perform the following tasks for all the questions listed in this section:

- **write** an **SQL query**,
- **execute** the query,
- **Check** your output **table** gives the expected result,
- **paste** the SQL **query** statement into this document.

- D1.** Output the average (mean) **standard\_cost** of all products from the **products** table, and name the output column as “Average Cost”.
- D2.** Output the average (mean) **list\_price** of all products from the **products** table, and give the output column the name of “Average List Price”.
- D3.** Output the number of product rows which have the category “Sauces” from the **products** table, and give the output column the name of “Sauce Count”.
- D4.** Output the highest **standard\_cost** of any item from the **products** table, and give the output column the name of “Highest Standard Cost”.
- D5.** From the **products** table, output the total number of **employees** in a column with the header “Number Of Employees”.

### More challenging questions:

- D6.** From the **products** table, output the average profit for all the products.  
Note: Profit can be calculated by finding the difference between the **list\_price** and the **standard\_cost**.
- D7.** From the **products** table, output the highest profit for any product.

## **Section E. SQL query using more complex WHERE statements**

Perform the following tasks for all the questions listed in this section:

- **write** an **SQL query**,
- **execute** the query,
- **Check** your output **table** gives the expected result,
- **paste** the **SQL query** statement into this document.

- E1.** Use two different methods, output the **product\_code** of all products, From the **products** table, that are in the “Sauces” category that also have a **standard\_cost** between 10 and 20 inclusive.
- E2.** We need German native speakers at our head office in the USA. Output the particulars of the employees, from the **employees** table, who currently live in the USA and have the word “German” contained within their notes or have a **country\_region** of “Germany”.
- E3.** Use two different methods, Output the **id** and **order\_id**, from the **invoices** table, for the invoices that are dated in the month of March in the year 2006.
- E4.** From the **inventory\_transactions** table, output the **id** and the **transaction\_type** of the transactions that were either:
- created in the month of March in 2006 and had a quantity above 100, or
  - created in the month of April in 2006 and had a quantity above 50.
- E5.** Find the relevant Entity table and output the **id** and **comments** of any **inventory\_transactions** that contains a comment (i.e., it is neither NULL nor an empty string).
- E6.** Find the relevant Entity table and output the full list of **order\_details** that either:
- Have a quantity greater than 20 and a **unit\_price** greater than 10, or
  - Have a **status\_id** of 1.



## **Section F. SQL query using **ORDER BY** and **LIMIT** statements**

Perform the following tasks for all the questions listed in this section:

- **write** an **SQL query**,
- **execute** the query,
- **Check** your output **table** gives the expected result,
- **paste** the **SQL query** statement into this document.

### **Note:**

**Although the questions may ask for a number (output X), there may **not be enough records** in which case **less than X will be output** – but the query should be correct.**

- F1.** Find the relevant Entity table, output the top 10 **order\_details** in terms of highest quantity.
- F2.** Find the relevant Entity table, output the 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> **order\_details** in terms of highest unit\_price.
- F3.** Find the relevant Entity table, output a list of all customers sorted alphabetically by (most important to least important): company, last\_name, first\_name.
- F4.** Find the relevant Entity table, output a list of the first five customers who are based (city) in Miami.
- F5.** Output from the **order\_details** table, the **product\_id** that has an **order\_id** value of 30 and has the lowest quantity.
- F6.** Find the relevant Entity table and sort the **customers** alphabetically by **company**, and output all details of the 5<sup>th</sup> to 10<sup>th</sup> customers on the list.
- F7.** Output from the **products** table, the details of the five highest **standard\_cost** products.