Mini world description: E-COMMERCE DATABASE

An e-commerce database stores data about customers, products, orders, deliveries, returns, reviews and wish lists (or baskets). The context of the database is as follows:

- Each customer is identified by an e-mail and has a name, birth date, phone number, password, address – building name or number (if any), street, city, country, and postcode. Customers can register different payment methods, such as credit/debit cards and vouchers/gift cards. For credit/debit cards, the database stores information on the card number, expiry date, verification code, and name on the card. It also registers whether a given card is the default payment method. Vouchers/gift cards are identified by serial numbers and have an expiry date, the total amount loaded in the card, and the current balance.

- Products are identified by a product number and have a short name (e.g., office chair), a textual description, brand, colour, dimensions (D x W x H), weight, price, number of days (or months/years) of warranty, and belong to a category (e.g., books, groceries, fashion, home & kitchen etc). The database monitors the stock of each product and makes a product "unavailable" when it is out of stock.

- Customers have baskets, and each basket has the list of products placed by the customer and the respective quantity of each product. There is no need to calculate and show the total of the basket as this is just a wish list.

- Each order placed by a customer is identified by an order number and has a date, the customer identification, one or more products (among those available in stock), subtotal for each product (this includes postage, VAT and other fees) based on the product and quantity being ordered, total (sum of subtotals), any deduction/promotion applied, and the grand total (the final price paid by the consumer). Each order also has the details of the payment method (for simplicity, we will assume only one payment method can be used in each order).

- Once an order is placed and paid, a delivery record is stored in the database. Each delivery is identified by a track number and has the order number, date of delivery, status (delivered, postponed, cancelled, pending) and the customer address.

- Customers can leave reviews on all products they ordered. Each review has a review number, date, customer identification, product name, text of the review and a ranking (from 1 to 5).

- The database also stores information about returns. Each return is identified by a ticket number, order number, start date (when the return was opened), due date (when the item should arrive back to the store), refund total (this should be based on the item being returned and its price – from the order – deducted by any return fee) and status (completed, cancelled, denied, pending).

Question 1 – Conceptual/E-R modelling: Design a conceptual/Entity-Relationship (E-R) model capturing the database context. Your model must have all the necessary entities and attributes, along with all the relationships and (min, max) cardinalities, according to the context. There is room for adjustments and/or extensions according to your understanding. Before moving to Question 2, make sure your conceptual model is entirely consistent and that no entities/attributes/relationships/cardinalities are missing.

Question 2 – Relational modelling: design a relational model based on your conceptual/E-R model. Ensure your relational model includes entities and attributes appearing on your conceptual model, along with all relationships. Make sure all cardinalities were properly transposed from your conceptual model (tip: revise the lecture materials and any other sources related to how one-to-one, one-to-many, and many-to-many relationships are mapped into relational schemas). Make sure all foreign keys are in place and that all attributes have a proper data type. Double-check your relational model before creating your database.

Question 3 – Database creation: create your database based on your relational model. Write (or generate from your relational model) DDL statements (i.e., CREATE TABLE...) to create all tables and relationships. You can decide whether a table will be a base table or a view, based on the data you are storing in each case. For instance, how to capture multiple orders from a single customer, how to put the list of products into a single order, or how to bring the delivery address into the delivery table? For any views, you may consider creating and populating the base tables first and then creating the necessary views. Remember that any views must appear on your relational model.

Question 4 – data loading: you must populate your database with enough data that allows you to answer all the subsequent questions (tip: read Question 5 carefully to check which data you need). The database must have a good list of products and distribution over the categories, a good number of customers – some of them with registered payment methods, a good number of orders – some of them from the same customer, as well as some products appearing in several orders, delivery records to all orders, and records for baskets, reviews, and returns. Ensure you have a good distribution of all status for deliveries and returns. You can write DML statements (i.e., INSERT INTO <TABLE> VALUES...) to populate data into your database, use any existing datasets, generate synthetic data or a mixture of all approaches. Ensure you have data compatible with your database model. See references.

Question 5 – SQL programming: write SQL statements to answer the following questions. You are free to explore any SQL constructs – join, subqueries, aggregation functions, user-defined functions (UDF), window functions etc.

5.1) You were asked to produce a list of all customers and their orders, including the list of products in each order and the (grand) total paid. Your query must show the customer's name and email, order number, order date, the list of products in each order and the total of each order.

5.2) You were asked to list all customers who have items in their baskets, so the company can make special offers based on their birthdays and any balance on existing gift cards. You must retrieve the customer identification (e-mail), name, birthday, any balance from gift cards, and the list of products in their baskets.

5.3) You were asked to identify the top two items sold in each product category, so the company can ensure that these products are kept in stock and marketed prominently. You must retrieve the category names, the top two products from each category, and their total sales.

5.4) You were asked to provide the month-over-month sales growth (i.e., the percentage change in sales from one month to the next), so the company can look at increasing and decreasing sales and identify trends. You must show the year, month, total sales and sales growth. Remember that any item returned and refunded (i.e., status = confirmed) must be taken into consideration when calculating the total sales of each month.

Question 6 – database consistency: write triggers to keep your database consistent. You must choose one situation where a trigger is necessary, such as (i) checking whether a product is in stock when the consumer places an order, (ii) the total of an order exceeds the gift card/voucher used as a payment method, (iii) the customer tries to open a new return when they already have a pending return, (iv) removing from the basket any items purchased by the customer, or any other specific case in your database. Write a trigger and demonstrate its use by (i) a query that passes the trigger (i.e., normal situation, no errors) and (ii) a query that violates any condition and "fires" the trigger (error message).

Presentation: good quality of all outputs (figures, code, report). Complete and organised report, addressing all requested sections/information. Good code documentation, when necessary. Complete set of references and compliance with all necessary policies.

Marking criteria

See the attached file.

Key dates

- Assignment release: 25/10/2024, 6 pm
- Solution deadline: 11/11/2024, 6 pm (both GitHub and Moodle)
- Feedback and provisional marks: 29/11/2024, 8 pm
- Requests for extension: 11/11/2024, noon
- Requests for revision of feedback and provisional marks: 13/12/2024, noon

---

Database tools

You can use any database modelling tools and/or software in this assignment. You can also write all your solutions in a Python notebook with the necessary database libraries. There is no preference for any tools, and no marks will be given if you use a more complex or simpler tool. Just make sure you can produce all the necessary outputs.

Generative AI tools

As per School and course-specific policy, you may acknowledge the use of any generative AI tool in any part of your summative work. You may note that marks can be deducted if no acknowledgement is made and/or a substantial part of your work (especially coding) is done by these tools. See Moodle for guidance.

You may use these tools literally as a "co-pilot" to help you prototype your database models, generate synthetic data, and/or structure your SQL queries, but the final results must be your own, validated work.

References

Kaggle or any other data sources

- https://www.kaggle.com/datasets/zusmani/pakistans-largest-ecommerce-dataset/data
- https://www.kaggle.com/datasets/thedevastator/unlock-profits-with-e-commerce-sales-data/data?select=Sale+Report.csv
- https://www.kaggle.com/datasets/carrie1/ecommerce-data/data?select=data.csv

Synthetic data generation

- https://www.datacamp.com/tutorial/creating-synthetic-data-with-python-faker-tutorial
- https://www.turing.com/kb/synthetic-data-generation-techniques#generating-synthetic-data-using-python-based-libraries
- https://github.com/statice/awesome-synthetic-data
- https://docs.sdv.dev/sdv and https://github.com/sdv-dev/SDV
- https://www.forbes.com/sites/bernardmarr/2024/08/29/20-generative-ai-tools-for-creating-synthetic-data/
- https://www.tonic.ai/lp/end-critical-bugs

You can prompt ChatGPT or Microsoft Copilot and ask for synthetic data creation. See the attached file for an example prompt and useful tips.

Deliverables

Remember to NOT include any identifiable information on your submission. Use your candidate number whenever necessary.

- **Source file and exported image** of your **conceptual/E-R model**.
- **Source file and exported image** of your **relational model**. Make sure both images are legible (good size/resolution and clear font types).
- **Database (data) file.**
- **Chat log** or **source file** used to **generate synthetic data** (if any).
- **Source file** of your **SQL statements.** This can be a Python program (if you have used it) or a text file (.sql) with all your commands. Make sure to document your code as necessary.
- **PDF report** (up to 8 pages, excluding references) with the following structure:

    - your candidate number at the top of the document.

    - up to 1,5 A4 pages in your report explaining your conceptual/E-R model, more specifically (i) use of specialisation (if any) and whether they represent partial or total specialisations; (ii) any assumptions related to cardinalities – for instance, the use of optional (zero) or mandatory (one) as minimum cardinality, and one or many as the maximum cardinality in a given relationship. This can be a bullet point list explaining each relationship and its cardinalities; (iii) use of NULL values – whether a particular attribute is required (mandatory) or not in a given entity; and (iv) any other aspects of your model that were adapted from the provided context. There's no need to include the model (image) in your report; this should be submitted separately.

    - up to 1,5 A4 pages in your report explaining your relational model, more specifically (i) any adaptations from the conceptual/E-R model – for instance, multivalued attributes mapped as tables, many-to-many relationships mapped as tables (and necessary attributes), composite attributes, composite primary keys etc; (ii) any views and how they are structured – which attributes form the view and from which base tables they come from; (iii) any constraints related to minimum and maximum cardinalities, or optional versus mandatory participation; and (iv) any other aspects of your relational model that were adapted from the conceptual/E-R model. There's no need to include the model (image) in your report; this should be submitted separately.

    - Up to 1 A4 page explaining any specific aspects of your database creation, such as the order in which all tables and views were created, any relaxation of foreign key constraints to allow the database to be populated with data etc. There's no need to include the DDL statements (CREATE TABLE…) as they will be submitted in a separate file.

    - Up to 1 A4 page explaining your data sources, if any. You must provide a clear explanation of each dataset used as a data source for your database and which data (attributes) were captured. You must provide links to all data sources used in your work. If you generate synthetic data, then you must comment on how each table was generated and any subsequent data processing you did to ensure the generated data was consistent/adherent to your model. You must also include any chat logs and/or source files as part of your deliverables.

- For each question (5.1 to 6), you must include (i) the question number, up to 2 paragraphs explaining your rationale to solve the question, and (iii) a screenshot of the results. Make sure the results are legible (good resolution/size and clear font types). **You must include the result of each query in your report; we won't execute your code to assert any results. The absence of results may lead to zero marks!**
- A statement on the use of any generative AI tools, as per School and course-specific policies. See Moodle for guidance.

Submission

This a 2-step submission process, on GitHub and then on Moodle:

- All files should be uploaded into your "assignment1" repository on GitHub. Observe that GitHub has a limit on the size of each file (see here). If you have any large files (for instance, your database file), include in your report a link to an external repository (for instance, Google Drive, Dropbox, WeTransfer) and make sure the file is accessible for download.
- **DO NOT** use any other GitHub repository, as we won't track your submission, and this will cause delays in assessing your work and providing feedback and provisional marks. When you are ready to submit, remove any unnecessary files on your "assignment1" repository – keep only the relevant deliverables.
- Copy the URL address of your GitHub repository, go to the submission portal on Moodle and paste it or include it on a text file. Make sure to press Submit to complete your assignment submission.
- **These two steps must be done by the deadline.**

Extension requests

You have the right to ask for an extension under some circumstances, as per LSE Extension Policy. Check here for guidance and the necessary documents. **Please note that any extensions must be requested and approved before the deadline, so do not leave this to the last minute as it may incur late penalties being applied to your work.** You should submit your requests to Dr. Christine Yuen (BSc Data Science programme director) and Steve Ellis (Undergraduate Programmes Manager) for analysis. See here for contact details, under Academic Faculty and Professional Service Staff, respectively. If you have adjustments in place and need an extension, follow the same procedure and contact your teachers for advice.

Feedback and provisional marks

Feedback and provisional marks will be provided in a markdown (.md) file in your “assignment1” repository by the expected date. Please, note that we do our best to provide you with relevant and meaningful feedback by the intended deadline, but **we reserve the right to delay any feedback while any extension requests are in place. You may also note that all marks are provisional and subject to changes to comply with School and departmental policies on mark distributions and as a result of external examiners and sub-board revisions.**

Revision of feedback and provisional marks

We advise you to read carefully your feedback file and get in touch with the teaching staff to discuss any points. We are available to revise any specific parts of your feedback when there is a justifiable reason for that. Please, **raise any points regarding your feedback and provisional marks up to two weeks after receiving them**. We may refrain from revising any parts of your feedback and provisional marks later in the academic year due to the internal flow/processing of marks across department and school sub-boards and external examiners.