**Basi di Dati (14AFQPL, 14AFQPI)**

**Anno Accademico 2024-2025**

**Politecnico di Torino**

# Testing di Large Language Model per Text2SQL

## Obiettivo

La finalità di questa esercitazione è quella di, fissato uno schema logico relazionale, generare coppie domande-risposte relative al task Text2SQL (ovvero dato un testo in linguaggio naturale formulare la corrispondente query SQL) associabili a pattern di difficoltà differenti, e di testare le abilità di tre differenti Large Language Model (LLM), confrontando soluzioni generate con diversi prompt con la soluzione attesa.

## Descrizione della base di dati

La base di dati denominata *CompanyCustomerOrders* raccoglie le informazioni degli ordini ricevuti da un'azienda ed è progettata per gestire e tenere traccia di ordini, prodotti ordinati, clienti e indirizzi.

**Schema logico relazionale:**
PRODUCTS(product_id, product_type_code, product_name, product_price*)
CUSTOMERS(customer_id, *address_id*, payment_method_code, customer_number*, customer_name, customer_field, customer_phone, customer_email)
ADDRESSES(address_id, address_details)
CUSTOMER_ORDERS(*order_id*, *customer_id*, order_date, order_status)
ORDER_ITEMS(*order_id*, *product_id*, order_item_id, order_quantity)

Le chiavi primarie sono sottolineate e le chiavi esterne sono in *corsivo*. L'asterisco indica i campi opzionali.

La tabella **Products** raccoglie le informazioni su ogni prodotto disponibile nell'azienda. Ogni prodotto è identificato da un codice univoco (`product_id`), dal tipo di prodotto (`product_type_code`), un nome descrittivo del prodotto (`product_name`) e un campo opzionale che indica il prezzo del prodotto (`product_price`).
La tabella **Customers** memorizza le informazioni di ogni cliente. Ogni cliente è identificato da un codice univoco (`customer_id`) ed è caratterizzato dal codice dell'indirizzo (`address_id`), un codice che specifica il metodo di pagamento preferito (`payment_method_code`) e il numero della partita IVA (se presente) (`customer_number`). Sono inoltre noti altri dati del cliente come il nome (`customer_name`), il settore di riferimento del cliente (`customer_field`), il telefono di contatto (`customer_phone`) e l'indirizzo e-mail (`customer_email`).
La tabella **Addresses** contiene gli indirizzi dei clienti. Ogni indirizzo ha un identificatore unico (`address_id`) e una descrizione testuale (`address_details`) che contiene informazioni specifiche sull'indirizzo quali, ad esempio, il nome della via, la città o il codice postale.

La tabella **Customer_Orders** memorizza ogni ordine effettuato dai clienti. Ogni ordine è identificato da un codice univoco (`order_id`) e contiene i dati del cliente che l'ha effettuato (`customer_id`). I dettagli dell'ordine includono inoltre la data in cui è stato effettuato (`order_date`) e lo stato attuale dell'ordine (`order_status`) ad esempio "In attesa" o "Completato".
Infine, la tabella **Order_Items** riporta i dettagli dei singoli articoli di ciascun ordine del cliente. In particolare, si riportano un numero progressivo per ogni voce dell'ordine (`order_item_id`), il riferimento all'ordine (`order_id`) e il prodotto ordinato (`product_id`). Inoltre, è memorizzata la quantità ordinata per il rispettivo prodotto (`order_quantity`).

# Svolgimento

Nel file Excel fornito in allegato al testo dell'esercitazione è contenuta la tabella con i seguenti campi da compilare e consegnare come risultato finale dell'esercitazione:

- **Pattern di difficoltà**: pattern caratterizzante la query proposta
- **Domanda**: query proposta in linguaggio naturale
- **Soluzione attesa**: soluzione attesa rispetto alla domanda proposta
- **Lingua**: lingua utilizzata per il prompt (italiano o inglese)
- **Prompt**: testo da inviare al LLM
- **Codestral/GPT-4o/Gemini**: colonne nelle quali inserire il codice SQL generato come risposta dai rispettivi LLM
- **Analisi Codestral/GPT-4o/Gemini**: commento dello studente alla soluzione proposta dai rispettivi LLM

Al fondo alla tabella è presente un campo nel quale lo studente può riportare un commento generale sulle principali risultanze sperimentali evidenziate tra cui, ad es.,

- Punti forti e punti deboli di ciascun LLM
- Confronti tra LLM
- Analisi e confronti tra prompt
- Confronto tra lingue

Durante il laboratorio a ciascun studente è richiesto di:

- Creare **una domanda per ogni tipologia di pattern** indicata nel file Excel.
- Scrivere la soluzione attesa alla query proposta.
- Formulare i prompt testuali da inviare all'LLM. Occorre formulare **un prompt diverso per ciascuna riga della tabella.** Le caratteristiche del prompt dipendono **dalla lingua** (italiano o inglese) e **dal tipo:**
    - **Zero-Shot** Learning (ZSL): al modello non viene fornito nessun esempio
    - **Few-Shot** Learning (FSL): vengono forniti K esempi al modello. Per questo laboratorio K=3, cioè dovranno essere forniti **3 esempi**.

  Nota: per alcuni pattern (1, 6-10) occorre generare sia prompt di tipo Zero-Shot che di tipo Few-Shot; per i restanti pattern basta generare i prompt di tipo Zero-Shot. Nei prompt devono essere anche forniti gli schemi logici delle tabelle.Per tradurre i prompt dall'italiano all'inglese potete procedere manualmente oppure utilizzare un tool di traduzione automatica (DeepL o Google Traduttore)
- Interrogare **i tre modelli (LLM)** utilizzando i prompt generati:
    - **Codestral:** https://chat.mistral.ai/chat (selezionare Codestral nel menù accanto alla barra della chat)
    - **GPT-4o:** https://chatgpt.com/
    - **Gemini:** https://gemini.google.com/app

- Inserire l'output fornito dai vari LLM all'interno del file Excel **seguendo obbligatoriamente** lo schema di tabella Excel già fornito (il pattern 0, colorato in grigio, è un esempio).
- **Analizzare** le risposte fornite dai modelli e riportare i relativi commenti nel file Excel.

Per poter usare i vari modelli, occorre registrarsi tramite la propria e-mail personale (potete usare l'indirizzo studenti fornito da PoliTo) o effettuare l'accesso tramite account terzi (e.g. Google).

Lo svolgimento del laboratorio prevede **due fasi distinte** ciascuna delle quali consente di ottenere i punti del primo Homework. Il punteggio totale massimo assegnabile al primo Homework è **1 punto, diviso equamente tra le due fasi**.

## Fase 1 - in laboratorio (0.5 punti)

- Seguendo le indicazioni dell'esercitatore di laboratorio, collegarsi alle interfacce Web di ciascun LLM e generare le coppie domande-risposte con i relativi prompt, soluzioni e commenti.
- Prima del termine dell'esercitazione caricare il file Excel compilato (anche solo parzialmente) sul Portale della Didattica (sezione "Elaborati").
    - N.B. Prima del caricamento il file Excel andrà obbligatoriamente ridenominato **sXXXXXX_Cognome_Nome_Text2SQL.xlsx**

      Nota: in caso di più cognomi o nomi, seguire lo stesso ordine indicato sulla vostra pagina personale del portale della didattica
- La consegna è ritenuta valida **se e solo se** il file Excel
    - è stato caricato con la denominazione corretta;
    - contiene almeno **cinque** domande-risposte e i relativi prompt, soluzioni, e commenti;
    - Data e orario di consegna non superano l'orario di conclusione del proprio slot di laboratorio.
- La **consegna è individuale**. L'eventuale svolgimento del laboratorio a coppie sullo stesso PC del laboratorio è ammesso, ma rimane obbligatoria la consegna individuale del file Excel compilato entro i termini suddetti. Nel caso il laboratorio fosse eseguito a coppie, occorre seguire la seguente denominazione per il caricamento del file:

  **sXXXXXX_Cognome_Nome_sXXXXXX_Cognome2_Nome2_ Text2SQL.xlsx**

## Fase 2 – completamento del lavoro (0.5 punti)

- Completare individualmente la parte residua del laboratorio, qualora non completato, e inserire anche i **commenti finali** nel campo in basso.
- **Entro 10 giorni dallo svolgimento del laboratorio** caricare il file Excel completo e aggiornato sul Portale della Didattica (sezione "Elaborati").
    - N.B. Prima del caricamento il file Excel andrà ridenominato **sXXXXXX_Cognome_Nome_Text2SQL_Final.xlsx**
- La consegna è ritenuta valida **se e solo se** il file Excel
    - è stato caricato con la denominazione corretta;
    - contiene tutte le domande-risposte, le soluzioni, i prompt richiesti e i commenti finali;
    - La data di consegna non supera i dieci giorni successivi alla data di svolgimento del laboratorio.
- La consegna finale, analogamente a quella intermedia, è **individuale** (anche nel caso in cui il laboratorio sia stato svolto a coppie sul medesimo PC di laboratorio).

Nel caso il laboratorio fosse eseguito a coppie, occorre seguire la seguente denominazione per il caricamento del file:

**sXXXXXX_Cognome_Nome_sXXXXXX_Cognome2_Nome2_ Text2SQL.xlsx**

# Note

La struttura del file Excel **non deve essere modificata.** Si può variare solo la larghezza di righe/colonne.
Nel caso di più Nomi/Cognomi, seguire lo stesso ordine indicato sulla vostra pagina personale sul portale della didattica.
Durante il caricamente usare il nome del file come descrizione (escluso il formato. E.g. **sXXXXXX_Cognome_Nome_sXXXXXX_Cognome2_Nome2_ Text2SQL**)

# Pattern

Le query proposte seguono livelli di difficoltà crescente, secondo i seguenti pattern:

1. **Selezione e proiezione:** la (vostra) soluzione alla domanda proposta deve contenere le clausole SELECT, FROM e WHERE, più eventualmente ORDER BY.

2. **Raggruppamento:** la (vostra) soluzione alla domanda proposta deve contenere la clausola GROUP BY (senza HAVING).

3. **Raggruppamento con condizione:** la (vostra) soluzione alla domanda proposta deve contenere le clausole GROUP BY e HAVING.

4. **Annidamento:** la (vostra) soluzione alla domanda proposta deve contenere una SELECT annidata (utilizzando IN, NOT IN, EXISTS o NOT EXISTS).

5. **Table Function:** la (vostra) soluzione alla domanda proposta deve contenere una TABLE FUNCTION.

6. **Annidamento + Raggruppamento:** la (vostra) soluzione alla domanda proposta deve contenere una SELECT annidata e la clausola GROUP BY (senza HAVING).

7. **Table Function + Raggruppamento:** la (vostra) soluzione alla domanda proposta deve contenere una TABLE FUNCTION e la clausola GROUP BY (senza HAVING).

8. **Annidamento + Table Function:** la (vostra) soluzione alla domanda proposta deve contenere una SELECT annidata ed una TABLE FUNCTION.

9. **Annidamento + Raggruppamento con condizione:** la (vostra) soluzione alla domanda proposta deve contenere una SELECT annidata e le clausole GROUP BY e HAVING.

10. **Table Function + Raggruppamento con condizione:** la (vostra) soluzione alla domanda proposta deve contenere una TABLE FUNCTION e le clausole GROUP BY e HAVING.

# Esempi di prompt

**LLM prompting con Zero-Shot Learning:**

“Sei un assistente AI per la risoluzione di query in linguaggio SQL partendo da una domanda testuale.
Ti verranno forniti in input lo schema logico delle tabelle da utilizzare e la domanda testuale.
Tu dovrai fornire come output solamente il codice SQL.

Tabelle:
PRODUCTS(product_id, product_type_code, product_name, product_price)
CUSTOMERS(customer_id, *address_id*, payment_method_code, customer_number, customer_name, customer_field, customer_phone, customer_email)
ADDRESSES(address_id, address_details)
CUSTOMER_ORDERS(*order_id*, *customer_id*, order_date, order_status)
ORDER_ITEMS(*order_id*, *product_id*, order_item_id, order_quantity)

Domanda:
Mostra il nome e l’indirizzo e-mail dei clienti che hanno fatto un acquisto il 10 gennaio 2024.

Codice SQL:”

**LLM Prompting con Few-Shot Learning:**

"Sei un assistente AI per la risoluzione di query in linguaggio SQL partendo da una domanda testuale.
Ti verranno forniti in input lo schema logico delle tabelle da utilizzare e la domanda testuale.
In aggiunta ti verranno forniti degli esempi che contengono il pattern di interesse per la risoluzione.
Tu dovrai fornire come output solamente il codice SQL.

Tabelle:
PRODUCTS(product_id, product_type_code, product_name, product_price)
CUSTOMERS(customer_id, *address_id*, payment_method_code, customer_number, customer_name, customer_field, customer_phone, customer_email)
ADDRESSES(address_id, address_details)
CUSTOMER_ORDERS(*order_id*, *customer_id*, order_date, order_status)
ORDER_ITEMS(*order_id*, *product_id*, order_item_id, order_quantity)

Esempi:
Domanda 1:
Mostra il nome e il numero di telefono dei clienti che risiedono a Londra

Codice SQL 1:
```
SELECT c.customer_name, c.customer_phone
FROM CUSTOMERS c
JOIN ADDRESSES a ON c.address_id = a.address_id
WHERE a.address_details LIKE '%Londra%';
```

Domanda 2:
Mostrare gli indirizzi e-mail e il numero di telefono dei clienti che hanno comprato un televisore.

Codice SQL 2:
```
SELECT c.customer_email, c.customer_phone
FROM CUSTOMERS c
JOIN CUSTOMER_ORDERS o ON c.customer_id = o.customer_id
JOIN ORDER_ITEMS oi ON o.order_id = oi.order_id
JOIN PRODUCTS p ON oi.product_id = p.product_id
WHERE p.product_name = 'televisore';
```

Domanda 3:
Mostra il nome dei clienti che hanno acquistato prodotti più cari di 100 euro

Codice SQL 3:
```
SELECT DISTINCT c.customer_name
FROM CUSTOMERS c
JOIN CUSTOMER_ORDERS o ON c.customer_id = o.customer_id
JOIN ORDER_ITEMS oi ON o.order_id = oi.order_id
JOIN PRODUCTS p ON oi.product_id = p.product_id
WHERE p.product_price > 100;
```

Domanda:
Mostra nome e indirizzo e-mail dei clienti che hanno fatto un acquisto il 10 gennaio 2024.

Codice SQL:"