

PROGETTO BASI DI DATI

STEFANO BASO – ARMANDO MIO

MAT. 1231758 – 1223862

- THE GAMEBOX FACTORY -

1. ABSTRACT

The Gamebox Factory, nuova catena di negozi di videogiochi con sede principale a Milano, intende realizzare un database per la gestione di tutte le sue attività. L'azienda ha vari negozi sparsi in tutta Italia e in ognuno di essi vengo venduti vari prodotti. Questi possono essere console, videogiochi e merchandise di varie tipologie. Riguardo a console e videogiochi vengono inoltre registrati i dati relativi al produttore.

I clienti hanno la possibilità di abbonarsi e ricevere una tessera per accumulare punti e ricevere varie offerte. I clienti acquistano in negozio tramite un ordine effettuato nel negozio oppure online. Quest'ultimo farà partire una spedizione associata all'indirizzo dell'abbonamento del cliente.

L'azienda inoltre, come vantaggio per gli abbonati, mette a disposizione una sala per la LanParty, ovvero offre uno spazio per giocare all'interno del negozio con varie postazioni per le console.

ANALISI DEI REQUISITI

Di ogni **negozio** si registra un codice identificativo, gli orari di apertura e chiusura, la data di apertura e l'eventuale data di chiusura (che rimane null se il negozio è attualmente aperto). Per l'**indirizzo** si memorizza la regione, la provincia, la via e il numero civico.

I **prodotti** hanno un codice univoco, il prezzo e la disponibilità; si dividono in **videogiochi**, **console** e **merchandise**. Dei videogiochi inoltre si registra il nome, il genere, il pegi, la valutazione e se è online o meno. Delle console si tiene traccia del modello, del colore e della memoria (500 gb o 1tb). Infine degli accessori la categoria a cui appartengono e il nome.

Delle console e dei videogiochi si memorizza il **produttore** con nome, capitale e nazionalità.

I **clienti** che comprano prodotti possono **abbonarsi** con relativa registrazione di CF, nome, cognome, email e data di nascita, inoltre una volta abbonati accumulano punti ad ogni acquisto. Le **tessere** ricevute sono di 4 livelli e sono associate alle **offerte**.

Gli **ordini** effettuati sono relativi al negozio **fisico** oppure **online** (con sconto del 3% solo online). Di questi viene fatta partire una **spedizione** in cui si tiene traccia del numero, del costo di spedizione, del corriere utilizzato e dell'arrivo previsto.

Le **LanParty** sono presenti in tutti i negozi e oltre al codice identificativo della sala hanno un omaggio (bibita) gratuito e una connessione ad Internet. Al suo interno ci sono varie **postazioni** divise per numero e associate ad una console.

2.1 OPERAZIONI TIPICHE

GIORNALIERE	
Registrazione di un nuovo abbonato	1100
Acquisto di un videogioco	2750
Registrazione di un nuovo ordine da spedire	1300
Inserimento di uno prodotto da vendere	15

ANNUALI	
Apertura/chiusura di un negozio	3
Registrazione nuovo modello di console	1
Aggiornamento offerte della tessera	12

2.2 GLOSSARIO DEI TERMINI

PEGI	Indice di età consigliata ufficiale per giocare al determinato videogioco
LAN PARTY	Sale fisiche in cui si può tramite abbonamento giocare alle varie console a disposizione
MERCHANDISE	Raggruppamento di tutte le altre categorie al di fuori di videogiochi e console che comprendono accessori, peluche, funko pop, action figure
FUNKO POP	Marca di statuette collezionabili di vari videogiochi, film o serie
ACTION FIGURE	Simili ai precedenti appena citati ma di varie dimensioni diverse e in materiale diverso

3. PROGETTAZIONE CONCETTUALE

3.1 LISTA ENTITA'

Nel database sono presenti 17 entità comprese le figlie nelle generalizzazioni. Tutti gli attributi elencati di seguito sono NOT NULL tranne nei casi in cui è espressamente elencato.

INDIRIZZO		
REGIONE	String	
PROVINCIA	String	Sigla a due lettere delle varie province
VIA	String	
CIVICO	Int	Numero a due cifre

NEGOZIO		
COD	String, <i>primary key</i>	
ORARIO_APERTURA	Time	Orario fisso settimanale
ORARIO_CHIUSURA	Time	Orario fisso settimanale
DATA_APERTURA	Date	Data inaugurazione del negozio
DATA_CHIUSURA	Date, null	Se null allora il negozio è attualmente aperto, se diverso da null allora chiuso definitivamente

Generalizzazione di PRODOTTO che contiene come figlie VIDEOGIOCO, MERCHANDISE, CONSOLE di tipo parziale.

Padre.PRODOTTO		
COD	String, <i>primary key</i>	
NOME	String	
PREZZO	Decimal	
DISPONIBILITA	Boolean	1 se disponibile, 0 se attualmente esaurito

Figlio.VIDEOGIOCO		
GENERE	String	Categoria del gioco
ONLINE	Boolean	1 se presenta l'accesso all'online, 0 altrimenti
PEGI	Int	Età consigliata per giocare
VALUTAZIONE	Decimal	Recensione da 0.0 a 10.0

Figlio.MERCHANDISE		
CATEGORIA	String	Macrocategoria a cui appartiene l'oggetto

Figlio.CONSOLE		
MEMORIA	String	
COLORE	String	

PRODUTTORE		
NOME	String, <i>primary key</i>	
NAZIONALITA	String	Nazionalità della sede principale dell'azienda
CAPITALE	Int	

Generalizzazione di ORDINE che contiene come figlie FISICO e ONLINE di tipo totale.

Padre.ORDINE		
COD	String, <i>primary key</i>	
DATA_ACQUISTO	Date	Data del pagamento
METODO_PAGAMENTO	String	Nome della carta utilizzata per il pagamento, contanti se si tratta di un ordine fisico in negozio

Figlio.FISICO		
---------------	--	--

Figlio.ONLINE		
SCONTO	Int	Valido solo per ordini online

SPEDIZIONE		
NUMERO	String, <i>primary key</i>	
CORRIERE	String	Nome del corriere
COSTO	Decimal	Costo del solo trasporto
ARRIVO_PREVISTO	Date	Data circa della consegna
MODALITA	String	Modalità di consegna standard o veloce, costo maggiore per la veloce

TESSERA		
LIVELLO	String, <i>primary key</i>	Suddivisi in 4 livelli per vantaggi
COLORE	String	Cambia in base al livello
PREZZO	Decimal	

Generalizzazione di CLIENTE che contiene un unico figlio ABBONATO di tipo parziale.

Padre.CLIENTE		
CF	String, <i>primary key</i>	
NOME	String	
COGNOME	String	
DATA_NASCITA	Date	
EMAIL	String	

Figlio.ABBONATO		
PUNTI	Int	Punti accumulabili nella tessera, non scadono mai
DATA_INIZIO	Date	Data di acquisto della tessera
DATA_FINE	Date	Data di scadenza, se non viene aggiornata allora il cliente non ha rinnovato

LAN_PARTY		
SALA	String, <i>primary key</i>	
OMAGGIO	String	Bibita gratis ricevuta (il campo contiene il nome della marca)
CONNESSIONE	String	Velocità di connessione megabyte per secondo

POSTAZIONE		
NUMERO	Int	Numero singolo della postazione a cui verrà associata una console

OFFERTA		
CODICE	String, <i>primary key</i>	
SCONTO	Int	Contiene un valore espresso in percentuale

3.2 LISTA RELAZIONI

NEGOZIO – INDIRIZZO

LUOGO (1,1)

- Ad ogni indirizzo corrisponde un solo negozio (1, 1)
- Ad un indirizzo non è detto che corrisponde al negozio (ci sono gli indirizzi anche dei clienti) (0, 1)

NEGOZIO – PRODOTTO

PRESSO (N, M)

- In un negozio sono presenti più prodotti (1, N)
- Lo stesso prodotto si può trovare in più negozi (1, N)

NEGOZIO – LAN_PARTY

AVERE (1, 1)

- Ogni negozio ha una sola sala Lan (1, 1)
- Ogni sala è associata ad un solo negozio (1, 1)

NEGOZIO – FISICO

ASSOCIARE (1, N)

- Un negozio può avere più di un ordine fisico effettuato (1, N)
- L'ordine fisico può essere associato ad un solo negozio (1, 1)

INDIRIZZO – SPEDIZIONE

INVIARE (1, N)

- Lo stesso indirizzo può ricevere più spedizioni o anche nessuna (1, N)
- Una spedizione è associata ad un solo indirizzo (1, 1)

INDIRIZZO – CLIENTE

ABITARE (1, 1)

- L'indirizzo è associato ad un solo cliente ma può essere anche di un negozio (0, 1)
- Il cliente abita in un solo indirizzo (1, 1)

PRODOTTO – CLIENTE – ORDINE

ACQUISTA (N, M, X)

- Lo stesso prodotto può essere acquistato da più clienti e in più ordini (1, N)
- Il cliente può fare più ordini e comprare più prodotti (1, N)
- L'ordine può contenere più prodotti (1, N)

VIDEOGIOCO – PRODUTTORE

SVILUPPARE (1, N)

- Lo stesso produttore sviluppa più videogiochi (1, N)
- I videogiochi sono sviluppati da un solo produttore (1, 1)

CONSOLE – PRODUTTORE

ASSEMBLARE (1, N)

- Lo stesso produttore sviluppa più console (1, N)

- Le console sono sviluppate da un solo produttore (1, N)

CONSOLE – POSTAZIONE UTILIZZARE (1, N)

- Una console è associata ad una sola postazione (1, 1)
- Una postazione può avere anche più console (1, N)

LAN_PARTY – POSTAZIONE CONTIENE (1, N)

- Una sala Lan contiene più postazioni (1, N)
- Le postazioni sono associate ad una sala (1, 1)

LAN_PARTY – ABBONATO ACCEDERE (N, M)

- Ad una sala Lan possono accedere più abbonati (1, N)
- Gli abbonati possono anche accedere a più sale (1, N)

TESSERA – ABBONATO RICEVERE (1, N)

- Un abbonato possiede una sola tessera (1, 1)
- Lo stesso tipo di tessera può essere associato a più abbonati (1, N)

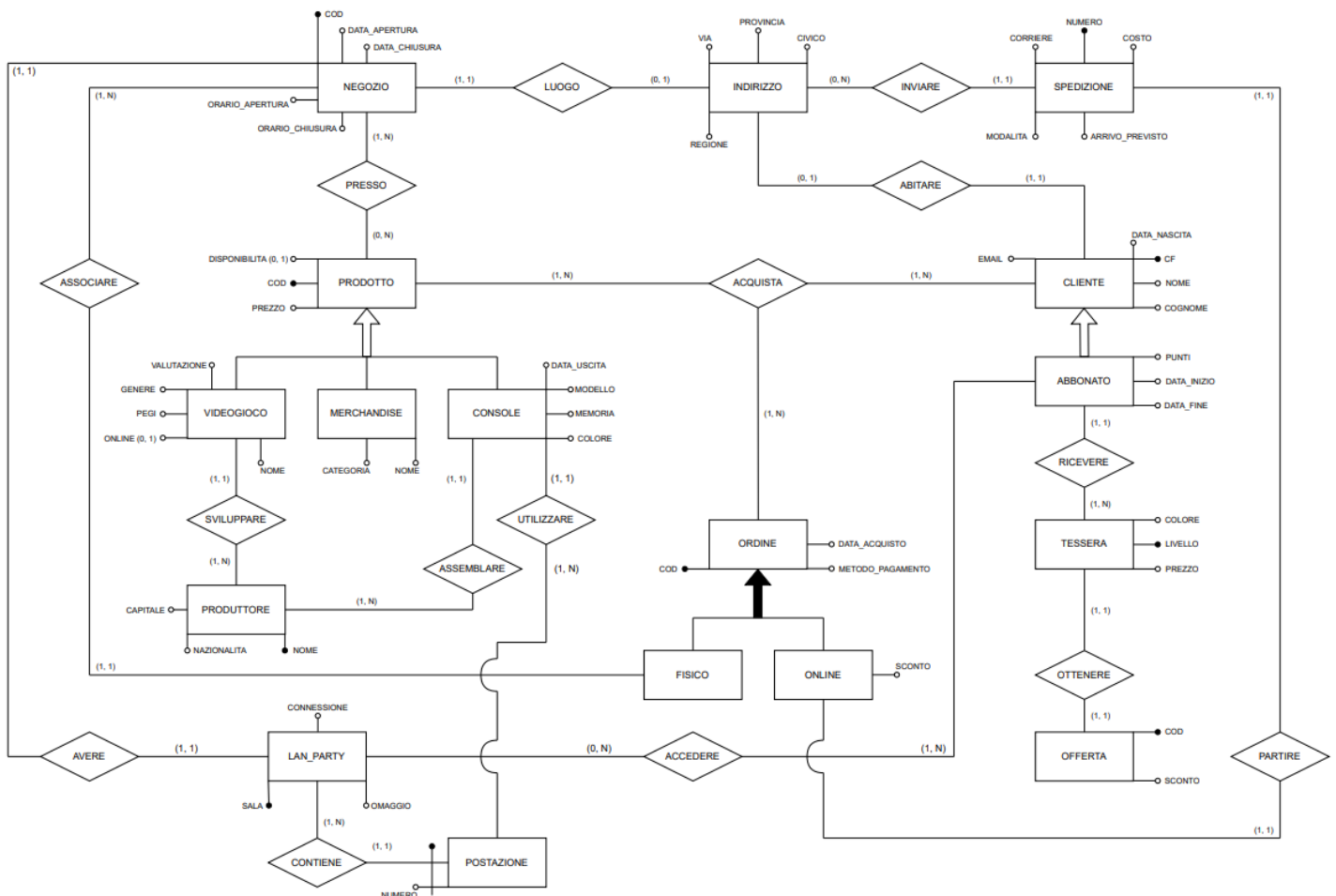
TESSERA – OFFERTE OTTENERE (1, 1)

- Una tessera ottiene una sola offerta che cambia nel tempo (1, 1)
- L'offerta è associata al tipo di tessera (1, 1)

ONLINE – SPEDIZIONE PARTIRE (1, N)

- Un ordine può far partire più spedizioni (1, N)
- La spedizione è associata ad un solo ordine (1, 1)

3.4 SCHEMA ENTITA' - RELAZIONE



4. ANALISI RIDONDANZE

Di seguito viene riportata l'analisi di due delle operazioni tipiche eseguite nel database.

OPERAZIONE 1: registrazione di un nuovo abbonato che accede ad una lan party (25 al giorno)

OPERAZIONE 2: visualizzare l'elenco delle spedizioni per controllo (750 al giorno)

CON RIDONDANZA

OPERAZIONE 1

CONCETTO	COSTRUTTO	ACCESSO	TIPO
LAN_PARTY	E	1	L
ACCEDERE	R	1	S

CLIENTE	E	1	L
CLIENTE	E	1	S

COSTI:

$25 \times 2 = 50$ accessi in lettura

$25 \times 2 = 50$ accessi in scrittura

OPERAZIONE 2

CONCETTO	COSTRUTTO	ACCESSO	TIPO
SPEDIZIONE	E	1	L

COSTI:

$750 \times 1 = 750$ accessi in lettura

TOTALE: $750 + 25 = 775$

SENZA RIDONDANZA

OPERAZIONE 1

CONCETTO	COSTRUTTO	ACCESSO	TIPO
LAN_PARTY	E	1	L
ACCEDERE	R	1	S
CLIENTE	E	1	L

COSTI:

$25 \times 2 = 50$ accessi in lettura

$25 \times 1 = 25$ accessi in scrittura

OPERAZIONE 2

CONCETTO	COSTRUTTO	ACCESSO	TIPO
SPEDIZIONE	E	1	L
PARTIRE	R	185	L

COSTI:

$186 \times 2 = 372$

TOTALE: $372 + 75 = 447$

In questo caso si ha un numero di accessi minore nel caso senza ridondanza

5. RISTRUTTURAZIONE SCHEMA ER

GENERALIZZAZIONE:

1 – *PADRE*: PRODOTTO

FIGLIE: VIDEOGIOCO, CONSOLE, MERCHANDISE

Risolto mantenendo sia padre che figlie, associando nuove chiavi primarie alle figlie e create le tre nuove relazioni collegate a PRODOTTO ovvero ESSERE_V, ESSERE_M, ESSERE_C.

2 – *PADRE*: ORDINE

FIGLIE: FISICO, NEGOZIO

Risolto eliminando le figlie tramite collasso verso l'alto e aggiunto l'attributo ONLINE_ che verrà popolato con 0 e 1. 0 se l'ordine è fisico mentre 1 per ordine online.

3 – *PADRE*: CLIENTE

FIGLIO: ABBONATO

Risolto eliminando il figlio tramite collasso verso l'alto e spostati gli attributi DATA_INIZIO, DATA_FINE e PUNTI su CLIENTE. Per differenziare gli abbonati attuali da chi non lo è più viene usato il campo DATA_FINE come risultato dell'ultimo rinnovo dell'abbonamento.

ATTRIBUTO BOOLEANO:

- DISPONIBILITA su PRODOTTO

Viene usato 1 se il prodotto è disponibile, 0 altrimenti.

- ONLINE su VIDEOGIOCO

Viene usato 1 se il videogioco richiede l'accesso al online, 0 altrimenti.

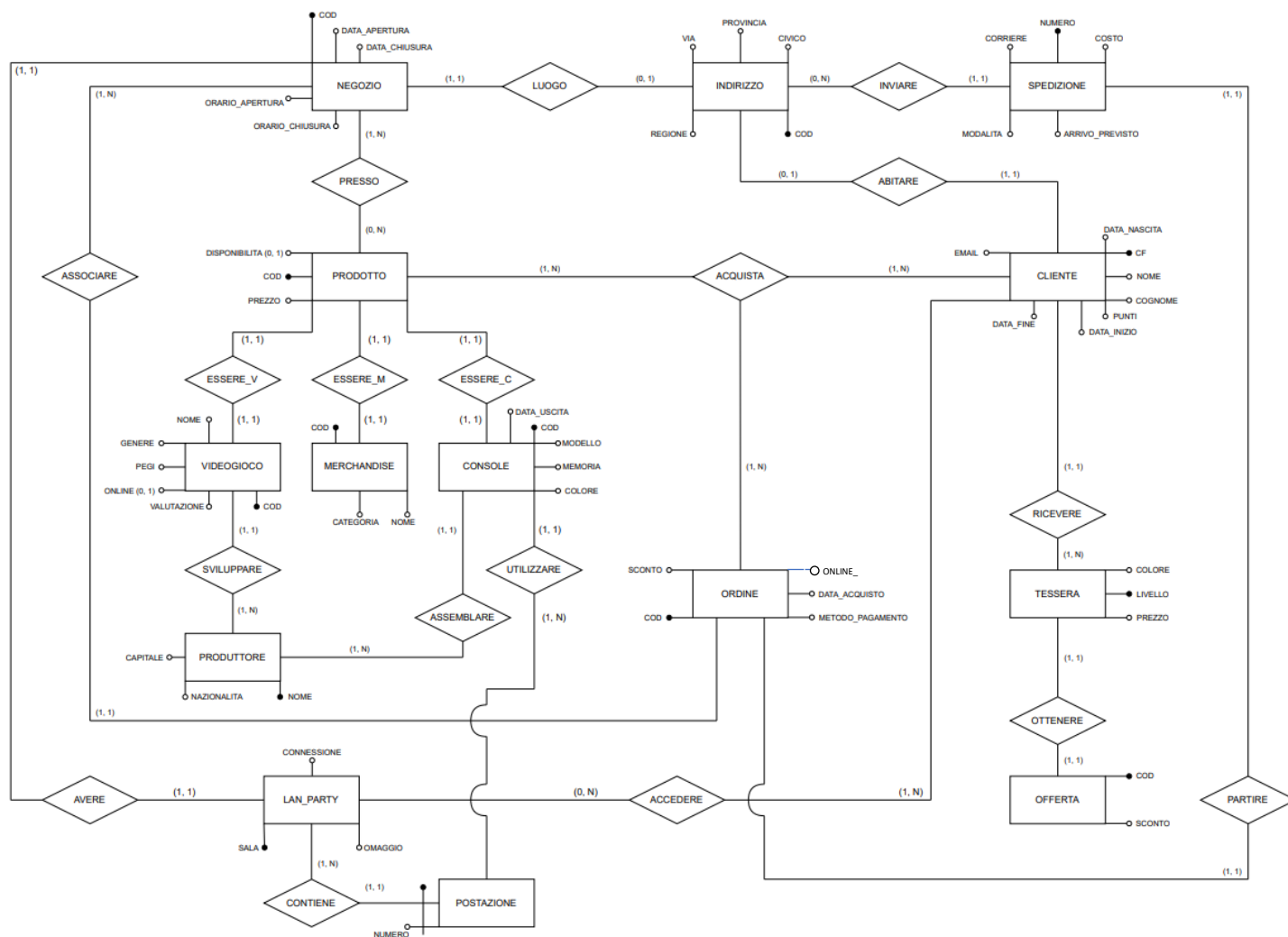
5.1 SCELTA NUOVE PRIMARY KEY

GENERALIZZAZIONE 1: creato l'attributo COD per tutte e tre le entità figlie VIDEOGIOCO, MERCHANDISE e CONSOLE. COD verrà usato come nuova chiave primaria.

GENERALIZZAZIONE 2: mantenuta la chiave primaria originale COD di ORDINE

GENERALIZZAZIONE 3: mantenuta la chiave primaria originale CF di CLIENTE

5.2 SCHEMA ER RISTRUTTURATO



5.3 SCHEMA TRADOTTO IN MODELLO RELAZIONALE

- $A \rightarrow \text{Tabella.B}$ indica il vincolo di chiave esterna

INDIRIZZO (COD, REGIONE, PROVINCIA, VIA, CIVICO)

NEGOZIO (COD, ORARIO_APERTURA, ORARIO_CHIUSURA, DATA_APERTURA, DATA_CHIUSURA, INDIRIZZO → INDIRIZZO.COD)

PRODOTTO	(COD, PREZZO, DISPONIBILITA, COD NEGOZIO → NEGOZIO.COD)
----------	---

PRODUTTORE (NOME, NAZIONALITA, CAPITALE)

VIDEOGIOCO (COD, NOME, GENERE, DATA_USCITA, PEGI, VALUTAZIONE, ONLINE_,
PRODUTTORE → PRODUTTORE.NOME, COD PRODOTTO → PRODOTTO.COD)

MERCHANDISE (COD, CATEGORIA, NOME, PRODUTTORE → PRODUTTORE.NOME,
COD PRODOTTO → PRODOTTO.COD)

CONSOLE	(<u>COD</u> , MODELLO, DATA_USCITA, COLORE, MEMORIA, PRODUTTORE → PRODUTTORE.NOME, COD_PRODOTTO → PRODOTTO.COD)
ORDINE	(<u>COD</u> , DATA_ACQUISTO, METODO_PAGAMENTO, SCONTO, ONLINE_, COD_NEGOZIO → NEGOZIO.COD)
SPEDIZIONE	(<u>NUMERO</u> , CORRIERE, COSTO, ARRIVO_PREVISTO, MODALITA, COD_ORDINE → ORDINE.COD, INDIRIZZO → INDIRIZZO.COD)
TESSERA	(<u>LIVELLO</u> , COLORE, PREZZO)
CLIENTE	(<u>CF</u> , NOME, COGNOME, EMAIL, DATA_NASCITA, ABBONATO, PUNTI, DATA_INIZIO, DATA_FINE, TESSERA → TESSERA.LIVELLO, INDIRIZZO → INDIRIZZO.COD)
LAN_PARTY	(<u>SALA</u> , OMAGGIO, CONNESSIONE, COD_NEGOZIO → NEGOZIO.COD)
POSTAZIONE	(<u>NUMERO</u> , SALA_LAN → LAN_PARTY.SALA, CONSOLE → CONSOLE.COD)
ACQUISTA	(<u>CF_CLIENTE</u> → CLIENTE.CF, <u>PRODOTTO</u> → PRODOTTO.COD, <u>CONSOLE</u> → CONSOLE.COD)
OFFERTA	(<u>COD</u> , SCONTO, <u>TESSERA</u> → TESSERA.LIVELLO)
ACCEDERE	(<u>SALA_LAN</u> → LAN_PARTY.SALA, <u>ABBONATO</u> → CLIENTE.CF)
PRESSO	(<u>COD_NEGOZIO</u> → NEGOZIO.COD, <u>COD_PRODOTTO</u> → PRODOTTO.COD)

6. QUERY, INDICE E C++

- **QUERY 1.** Lista dei primi 10 clienti che hanno acquistato di più online

```
1008 SELECT ACQUISTA.CF_CLIENTE, SUM(CASE WHEN ORDINE.ONLINE_='1' THEN 1 ELSE 0 END) AS TOT_ORDINI
1009 FROM ACQUISTA INNER JOIN ORDINE ON ACQUISTA.ORDINE=ORDINE.COD
1010 GROUP BY CF_CLIENTE
1011 ORDER BY TOT_ORDINI DESC
1012 LIMIT 10
1013
```

cf_cliente	tot_ordini
MOKC94	2
MGSR99	2
GLDS96	1
ERSV95	1
ÉNSA96	1
AYMS97	1
HNLE90	1
DOGS97	1
MESR99	1
SOOA92	1

- **QUERY 2.** Bilancio del primo semestre (gennaio – giugno) con numero di ordini effettuati e guadagno

```
1016 SELECT COUNT(ACQUISTA.ORDINE) AS TOT_ORDINI, SUM(PRODOTTO.PREZZO) AS LORDO
1017 FROM PRODOTTO INNER JOIN ACQUISTA ON PRODOTTO.COD=ACQUISTA.PRODOTTO
1018         INNER JOIN ORDINE ON ORDINE.COD=ACQUISTA.ORDINE
1019 WHERE PRODOTTO.COD=ACQUISTA.PRODOTTO
1020        AND ORDINE.DATA_ACQUISTO BETWEEN '2022-01-01' AND '2022-06-30'
1021
```

tot_ordini	lordo
35	5637.58

- **QUERY 3.** Tipologie di videogiochi acquistati, numero di ordini per tipologia e incassi

```

1025 SELECT VIDEOGIOCO.GENERE, COUNT(ACQUISTA.ORDINE) AS GIOCHI_VENDUTI, SUM(PRODOTTO.PREZZO) AS INCASSO
1026 FROM ACQUISTA INNER JOIN PRODOTTO ON PRODOTTO.COD=ACQUISTA.PRODOTTO
1027         INNER JOIN VIDEOGIOCO ON PRODOTTO.COD=VIDEOGIOCO.COD_PRODOTTO
1028 GROUP BY VIDEOGIOCO.GENERE

```

genre	giochi_venduti	incasso
Adventure	2	78.00
MOBA	2	148.00
Open World	1	49.00
Hack and Slash	2	98.00

- **QUERY 4.** numero di abbonati lv3 che accedono alle varie sale lan

```

1007 SELECT ACCEDERE.SALA_LAN, COUNT(ACCEDERE.ABBONATO) AS NUMERO_ABBONATI
1008 FROM ACCEDERE INNER JOIN CLIENTE ON ACCEDERE.ABBONATO=CLIENTE.CF
1009 WHERE CLIENTE.TESSERA='LV3'
1010 GROUP BY ACCEDERE.SALA_LAN
1011 ORDER BY NUMERO_ABBONATI DESC

```

sala_lan	numero_abbonati
NA0206001	3
VE0226001	2
MI0327001	1
MI0401001	1
NA0115001	1
PD0228001	1

- **QUERY 5.** quanto pagano i clienti che accedono alla lan_party divisi per omaggi offerti

```

1006 SELECT LAN_PARTY.OMAGGIO, COUNT(ACCEDERE.ABBONATO) AS TOT_PRESENTI, SUM(TESSERA.PREZZO) AS INCASSI_DA_TESSERA
1007 FROM LAN_PARTY INNER JOIN ACCEDERE ON LAN_PARTY.SALA=ACCEDERE.SALA_LAN
1008             INNER JOIN CLIENTE ON ACCEDERE.ABBONATO=CLIENTE.CF
1009             INNER JOIN TESSERA ON CLIENTE.TESSERA=TESSERA.LIVELLO
1010 GROUP BY LAN_PARTY.OMAGGIO

```

omaggio	tot_presenti	incassi_da_tessera
Coca Cola	10	134.90
RedBull	8	84.92
Sprite	4	44.96
Pepsi	10	104.90
Fanta	12	174.88

- **QUERY 6.** lista negozi con ordini fisici

```

1044 SELECT DISTINCT NEGOZIO.COD, COUNT(ORDINE.COD) AS ORDINI_FISICI
1045 FROM NEGOZIO INNER JOIN ORDINE ON NEGOZIO.COD=ORDINE.COD_NEGOZIO
1046 WHERE ORDINE.ONLINE_='0'
1047         AND NEGOZIO.COD=ORDINE.COD_NEGOZIO
1048 GROUP BY NEGOZIO.COD

```

cod	ordini_fisici
FI0122	1
FI0304	1
FI0519	1
MI0203	1
MI0327	1
MI0401	1
NA0115	1
NA0206	1
PD0228	1
RM0218	1
RM0519	1
VE0226	1

6.1 INDICE

Per una visualizzazione più rapida della verifica della disponibilità dei prodotti si richiede la creazione di un indice per tale scopo.

```
CREATE INDEX DISPONIBILITA_PRODOTTI ON PRODOTTO (COD_PRODOTTO, DISPONIBILITA);
```

6.2 CODICE C++

Il codice c per l'esecuzione delle query viene gestito all'interno del main con all'inizio la connessione al database tramite il costrutto TRY e CATCH e controllo di connessione con eventuale errore di disconnessione.

- prepareStatement() e executeQuery() vengono usati per il comando di selezione delle Query.
- next() viene usato per passare ai record dei risultati della selezione.
- getInt() e getString() vengono usati per analizzare il tipo di risultati.

NOTA: le query sono su un unico file ma dopo ogni query eseguita correttamente il sistema si mette in pausa
