15420765101 11454232046 27401706422 75721021601 14307131652 43135317123 134115447451

COGNOME	NOME	MATRICOLA

Basi di Dati 2022/23 – 18 luglio 2023

Closed book (non è possibile consultare materiale)

Tempo a disposizione: 1h 45' parte I e II [1h 20' se senza esercizio I.A] + 45' parte III

Esercizio I.A REVERSE ENGINEERING * gli studenti che hanno aderito a opzione 2 sono esonerati

Si consideri il seguente schema relazionale, relativo ad assicurazioni sanitarie e alle relative pratiche per il rimborso delle prestazioni coperte dall'assicurazione.

ASSISTITO(CodiceFisc, Nome, Cognome, Genere, DataN)
TITOLARE(Titolare^{ASSISTITO}, *Email*, Telefono, Indirizzo, Convenzione$_O$)
FAMILIARE(Familiare^{ASSISTITO}, Titolare^{TITOLARE})
POLIZZA(NumP, Titolare^{TITOLARE}, DataIn, DataScad, Tipo, ImportoP)
COPRE_FAM(NumP^{POLIZZA}, Familiare^{FAMILIARE})
SOGGETTO(CodiceSogg, Denominazione, Indirizzo)
PRATICA(CodicePr, NumP^{POLIZZA}, DataPratica)
PRESTAZIONE(CodicePr^{PRATICA}, NumRiga, Erogante^{SOGGETTO}, Beneficiario^{ASSISTITO}, Descrizione, DataPrest, ImportoPrest)

1. si proponga uno schema concettuale Entity Relationship la cui traduzione dia luogo a tale schema logico

2. si modifichi lo schema in 1. per gestire il fatto che un soggetto possa essere una persona fisica (medico specialista/infermiere/…) oppure una società (laboratorio di analisi, …), con in un caso codice fiscale e la qualifica con eventuali specializzazioni e nell'altro la ragione sociale e il tipo di società.

COGNOME	NOME	MATRICOLA

Esercizio I.B NORMALIZZAZIONE

1. Si consideri il seguente schema di relazione, che rappresenta alcune informazioni sugli spot trasmessi giornalmente su alcuni canali. Il tipo di un canale può essere televisivo o radiofonico.

   SPOT(NomeCanale, TipoCanale, Lunghezza, Prodotto, Categoria, Ora)

   Determinare, per ciascuna delle seguenti affermazioni, se rappresentano dipendenze funzionali per la relazione SPOT. In caso affermativo, presentare la dipendenza; in caso negativo, motivare l'affermazione.

   a) Ogni spot può essere trasmesso su canali diversi.

   b) Gli spot per un certo prodotto vengono trasmessi in una certa ora sempre con la stessa lunghezza.

   c) La categoria di un prodotto è unica.

2. Si considerino lo schema di relazione R(A,B,C,D,E,F) e l'insieme di dipendenze funzionali G = {A→B, C→AD, AF→EC}. Si determinino le chiavi candidate di R. Si stabilisca se R è in 3NF. Qualora non lo sia, si definisca una decomposizione di R in 3NF che conservi le dipendenze date. Si determini se la decomposizione risultante è in BCNF. Motivare le risposte.

COGNOME	NOME	MATRICOLA

Esercizio II.A – ALGEBRA RELAZIONALE

In riferimento al seguente schema relazionale, relativo ad assicurazioni sanitarie e alle relative pratiche per il rimborso delle prestazioni coperte dall'assicurazione

ASSISTITO(CodiceFisc, Nome, Cognome, Genere, DataN)
TITOLARE(Titolare^{ASSISTITO}, *Email*, Telefono, Indirizzo, Convenzione$_O$)
FAMILIARE(Familiare^{ASSISTITO}, Titolare^{TITOLARE})
POLIZZA(NumP, Titolare^{TITOLARE}, DataIn, DataScad, Tipo, ImportoP)
COPRE_FAM(NumP^{POLIZZA}, Familiare^{FAMILIARE})
SOGGETTO(CodiceSogg, Denominazione, Indirizzo)
PRATICA(CodicePr, NumP^{POLIZZA}, DataPratica)
PRESTAZIONE(CodicePr^{PRATICA}, NumRiga, Erogante^{SOGGETTO}, Beneficiario^{ASSISTITO}, Descrizione, DataPrest, ImportoPrest)

Formulare le seguenti interrogazioni in **algebra relazionale**.
1. Determinare nome e cognome delle donne titolari di polizze in corso che coprono almeno un loro familiare.

2. Determinare l'email del titolare e la data di scadenza delle polizze per cui sono state presentate pratiche che includono sia una prestazione di tipo "visita ortopedica" che una prestazione di tipo "risonanza magnetica ginocchio" (valori per attributo Descrizione in PRESTAZIONE)

Suggerimento per verifica/autovalutazione: Per ogni interrogazione, dopo averla formulata, effettuare i controlli richiesti e validare con V se si ritiene che il controllo sia superato, con X se si ritiene che non lo sia.

Verifica/autovalutazione	*a)*	*b)*
L'interrogazione formulata è corretta dal punto di vista dei vincoli di schema		
La richiesta e l'interrogazione formulata restituiscono una relazione con lo stesso schema		
La richiesta e l'interrogazione formulata sono entrambe monotone/non monotone		
Su una piccola istanza, la richiesta e l'interrogazione formulata restituiscono lo stesso risultato		

COGNOME	NOME	MATRICOLA

Esercizio II.B - SQL

In riferimento al seguente schema relazionale, relativo ad assicurazioni sanitarie e alle relative pratiche per il rimborso delle prestazioni coperte dall'assicurazione

ASSISTITO(CodiceFisc, Nome, Cognome, Genere, DataN)
TITOLARE(Titolare^{ASSISTITO}, *Email*, Telefono, Indirizzo, Convenzione$_O$)
FAMILIARE(Familiare^{ASSISTITO}, Titolare^{TITOLARE})
POLIZZA(NumP, Titolare^{TITOLARE}, DataIn, DataScad, Tipo, ImportoP)
COPRE_FAM(NumP^{POLIZZA}, Familiare^{FAMILIARE})
SOGGETTO(CodiceSogg, Denominazione, Indirizzo)
PRATICA(CodicePr, NumP^{POLIZZA}, DataPratica)
PRESTAZIONE(CodicePr^{PRATICA}, NumRiga, Erogante^{SOGGETTO}, Beneficiario^{ASSISTITO}, Descrizione, DataPrest, ImportoPrest)

Formulare le seguenti interrogazioni in **SQL**.
1. Determinare l'email e il numero di telefono del titolare insieme al numero di polizza per le polizze scadute (=data di scadenza precedente alla data odierna) per cui non sono state presentate pratiche.

2. Determinare le polizze il cui importo è inferiore all'importo delle prestazioni coperte da tale polizza.

15420765101 11454232046 27401706422 75721021601 14307131652 43135317123 134115447451

COGNOME	NOME	MATRICOLA

Op.	Funzionalità	Cond.	Semantica
Π_A	$\mathcal{R}(U) \to \mathcal{R}(A)$	$A \subseteq U$	$\Pi_A(R) = \{t[A] \mid t \in R\}$
σ_F	$\mathcal{R}(U) \to \mathcal{R}(U)$	$A(F) \subseteq U$	$\sigma_F(R) = \{t \mid t \in R \wedge F(t)\}$
$\times$	$\mathcal{R}(U) \times \mathcal{R}(V) \to \mathcal{R}(U \cup V)$	$U \cap V = \emptyset$	$R_1 \times R_2 = \{t_1 \cdot t_2 \mid t_1 \in R_1 \wedge t_2 \in R_2\}$
$\cup$	$\mathcal{R}(U) \times \mathcal{R}(U) \to \mathcal{R}(U)$		$R_1 \cup R_2 = \{t \mid t \in R_1 \vee t \in R_2\}$
$-$	$\mathcal{R}(U) \times \mathcal{R}(U) \to \mathcal{R}(U)$		$R_1 - R_2 = \{t \mid t \in R_1 \wedge t \notin R_2\}$
$\cap$	$\mathcal{R}(U) \times \mathcal{R}(U) \to \mathcal{R}(U)$		$R_1 \cap R_2 = \{t \mid t \in R_1 \wedge t \in R_2\}$
$\bowtie_F$	$\mathcal{R}(U) \times \mathcal{R}(V) \to \mathcal{R}(U \cup V)$	$U \cap V = \emptyset$	$R_1 \bowtie_F R_2 = \{t_1 \cdot t_2 \mid t_1 \in R_1 \wedge t_2 \in R_2 \wedge F(t_1 \cdot t_2)\}$
$\bowtie$	$\mathcal{R}(U) \times \mathcal{R}(V) \to \mathcal{R}(U \cup V)$		$R_1 \bowtie R_2 = \{t \mid t[U] \in R_1 \wedge t[V] \in R_2\}$
$\div$	$\mathcal{R}(U) \times \mathcal{R}(V) \to \mathcal{R}(U \setminus V)$	$V \subset U$	$R_1 \div R_2 = \{t \mid \forall t_2 \in R_2 \, \exists t_1 \in R_1 \; t.c. \; t_1[U \setminus V] = t, t_1[V] = t_2\}$

$\rho_{A \leftarrow A'}$ $\quad R(U) \to R(U \setminus A \cup A') \quad A \subseteq U$

SQL

SELECT *column_name(s)* **FROM** *table_name*	Select data from a table.
SELECT * FROM *table_name*	Select all data from a table.
SELECT DISTINCT *column_name(s)* **FROM** *table_name*	Select only distinct (different) data from a table.
SELECT *column_name(s)* **FROM** *table_name* **WHERE** *column operator value* **AND** *column operator value* **OR** *column operator value* **AND (... OR ...)** **...**	Select only certain data from a table.

Operato

Operator	
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern. A "%" sign can be used to define wildcards (missing letters in the pattern)

SELECT *column_name(s)* **FROM** *table_name* **WHERE** *column_name* **IN (***value1***,** *value2***, ...)**	The IN operator may be used if you know the exact value you want to return for at least one of the columns.
SELECT *column_name(s)* **FROM** *table_name* **ORDER BY** *row_1***,** *row_2* **DESC,** *row_3* **ASC, ...**	Select data from a table with sort the rows. **ASC** (ascend) is a alphabetical and numerical order (optional) **DESC** (descend) is a reverse alphabetical and numerical order
SELECT *column_1***, ...,** ***AGGREGATE_FUN*****(***agg_column_name***)** **FROM** *table_name* **GROUP BY** *group_column_name*	Without the GROUP BY all the tuples in the table are gouped together

ome aggregate

Function	
AVG(column)	Returns the average value of a column
COUNT(column)	Returns the number of rows (without a NULL value) of a column
MAX(column)	Returns the highest value of a column
MIN(column)	Returns the lowest value of a column
SUM(column)	Returns the total sum of a column

SELECT *column_1***, ...,** ***AGGREGATE_FUN*****(***agg_column_name***)** **FROM** *table_name* **GROUP BY** *group_column_name* **HAVING** **SUM(***group_column_name***)** *condition value*	HAVING... was added to SQL because the WHERE keyword could not be used against aggregate functions (like SUM), and without HAVING... it would be impossible to test for result conditions.
SELECT *column_name* **AS** *column_alias* **FROM** *table_name*	Column name alias
SELECT *table_alias.column_name* **FROM** *table_name* **AS** *table_alias*	Table name alias
SELECT *column_1_name***,** *column_2_name***, ...** **FROM** *first_table_name* **JOIN** *second_table_name* **ON** *first_table_name.keyfield* **=** *second_table_name.foreign_keyfield*	The (INNER) JOIN returns all rows from both tables where there is a match. If there are rows in first table that do not have matches in second table, those rows will not be listed. If this is not the intended behavior, use OUTER JOIN instead (RIGHT/LEFT/FULL)
SQL_Statement_1 **UNION** *SQL_Statement_2*	Select all different values from *SQL_Statement_1* and *SQL_Statement_2*

COGNOME	NOME	MATRICOLA

PARTE III. DOMANDE, SOLO PER 12 CFU

a) Descrivere il concetto di piano di esecuzione logico, presentandone anche un esempio.

b) Discutere i pro e i contro degli indici ordinati multi-attributo.

c) Discutere vantaggi e svantaggi nell'uso dei ruoli per il controllo dell'accesso in SQL.