

UNIVERSITÀ DELLA CALABRIA

Facoltà di SS.MM.FF.NN.

Corso di Laurea in Informatica

A.A. 2019/2020

PROGETTO PER IL CORSO DI

BASI DI DATI RELAZIONALI

DOCENTE: PROF. P. RULLO

**LABORATORIO: ING. G. LABOCETTA,
DOTT.SSA D. ANGILICA**

SISTEMA INFORMATIVO

PER LA GESTIONE DI

UN ISTITUTO COMPRENSIVO.

GRUPPO 26

<200856 Salvatore Gatto>

<200864 Davide Ragona>

<201064 Giorgio Francesco Guagliardi>

1. TEMATICA PROGETTUALE

La progettazione del sistema informatico in esame riguarda la gestione di un istituto comprensivo.

2. RACCOLTA E ANALISI DEI REQUISITI

2.1 Raccolta dei requisiti

Schema dei requisiti richiesti dall'istituto comprensivo StudioOnline.

	REQUISITI RICHIESTI
1	L'istituto comprensivo StudioOnline vuole riorganizzare il proprio database a seguito
2	di alcuni aggiornamenti nella normativa.
3	Il complesso scolastico include tre gradi: scuola dell'infanzia, scuola primaria e scuola
4	secondaria inferiore. Ciascuna scuola è distribuita in diversi plessi e ciascun plesso
5	ospita anche più di un grado scolastico.
6	Le iscrizioni avvengono nel mese di gennaio: i genitori iscrivono i propri figli e di
7	ognuno di loro viene registrata l'anagrafica.
8	Ogni studente, in un anno scolastico (01/09-19/06), è iscritto ad uno dei 3 gradi
9	dell'istituto ma si vuole memorizzare per ciascun ragazzo lo storico delle iscrizioni.
10	All'atto dell'iscrizione, ogni studente è associato ad una classe.
11	I plessi sono identificati da un indirizzo e sono composti da delle aule. Di ogni aula si
12	conoscono i metri quadri, hanno un numero massimo di studenti ospitabili e la fascia
13	di età degli studenti ospitabili.
14	Ogni classe, che ogni anno può cambiare aula, è identificata da un numero (1-4 per
15	l'infanzia, 1-5 per la primaria e 1-3 per le medie) e da una lettera dell'alfabeto (aule
16	di gradi diversi possono avere la stessa coppia). Di ciascun insegnante si conosce,
17	ogni anno, il numero di ore lavorative settimanali e le classi a cui è assegnato.

2.2 Analisi dei requisiti

In questo paragrafo inseriremo la tabella guida con l'eliminazione delle ambiguità e in seguito la tabella dei prerequisiti richiesti ristrutturata secondo le linee guida della prima tabella.

2.2.1 Eliminazione delle Ambiguità

Linea	Termine	Sinonimi	Motivazione correzione
3	Istituto comprensivo	Complesso scolastico	L'istituto comprensivo e il complesso scolastico rappresentano entrambe la medesima entità.
9	Studente	Ragazzo	Il termine ragazzo è un doppiante del termine studente.
15	Secondaria inferiore	Medie	Il termine medie rende incoerente l'utilizzo degli aggettivi indicati per indicare i vari gradi dell'Istituto comprensivo.

2.2.2 Ristrutturazione dei requisiti richiesti

	REQUISITI RICHIESTI
1	L'istituto comprensivo StudioOnline vuole riorganizzare il proprio database a seguito
2	di alcuni aggiornamenti nella normativa.
3	L'istituto comprensivo include tre gradi: scuola dell'infanzia, scuola primaria e scuola
4	secondaria inferiore. Ciascuna scuola è distribuita in diversi plessi e ciascun plesso
5	ospita anche più di un grado scolastico.
6	Le iscrizioni avvengono nel mese di gennaio: i genitori iscrivono i propri figli e di
7	ognuno di loro viene registrata l'anagrafica.
8	Ogni studente, in un anno scolastico (01/09-19/06), è iscritto ad uno dei 3 gradi
9	dell'istituto ma si vuole memorizzare per ciascun di loro lo storico delle iscrizioni.
10	All'atto dell'iscrizione, ogni studente è associato ad una classe.
11	I plessi sono identificati da un indirizzo e sono composti da delle aule. Di ogni aula si
12	conoscono i metri quadri, hanno un numero massimo di studenti ospitabili e la fascia
13	di età degli studenti ospitabili.
14	Ogni classe, che ogni anno può cambiare aula, è identificata da un numero (1-4 per
15	l'infanzia, 1-5 per la primaria e 1-3 per le secondarie inferiori) e da una lettera
16	dell'alfabeto (aule di gradi diversi possono avere la stessa coppia). Di ciascun
17	insegnante si conosce, ogni anno, il numero di ore lavorative settimanali e le classi a cui è assegnato.

2.2.3 Raffinamento delle Specifiche e Individuazione dei Concetti di Base

Inserimento di una tabella con frasi di carattere generale utili alla comprensione della progettazione dello schema ER.

	FRASI DI CARATTERE GENERALE
1	L'istituto comprensivo StudioOnline vuole riorganizzare il proprio database a seguito di alcuni aggiornamenti nella normativa.

Inserimento di tabelle relative ad ogni entità con relative descrizioni.

FRASI RELATIVE A	
L'istituto comprensivo	L'istituto comprensivo include tre gradi: scuola dell'infanzia, scuola primaria e scuola secondaria inferiore.
Scuola	L'istituto comprensivo include tre gradi: scuola dell'infanzia, scuola primaria e scuola secondaria inferiore.
	Ciascuna scuola è distribuita in diversi plessi e ciascun plesso ospita anche più di un grado scolastico
Plesso	I plessi sono identificati da un indirizzo e sono composti da delle aule.
	Ciascun plesso ospita anche più di un grado scolastico.
Studente	Ogni studente, in un anno scolastico (01/09-19/06), è iscritto ad uno dei 3 gradi
	All'atto dell'iscrizione, ogni studente è associato ad una classe.

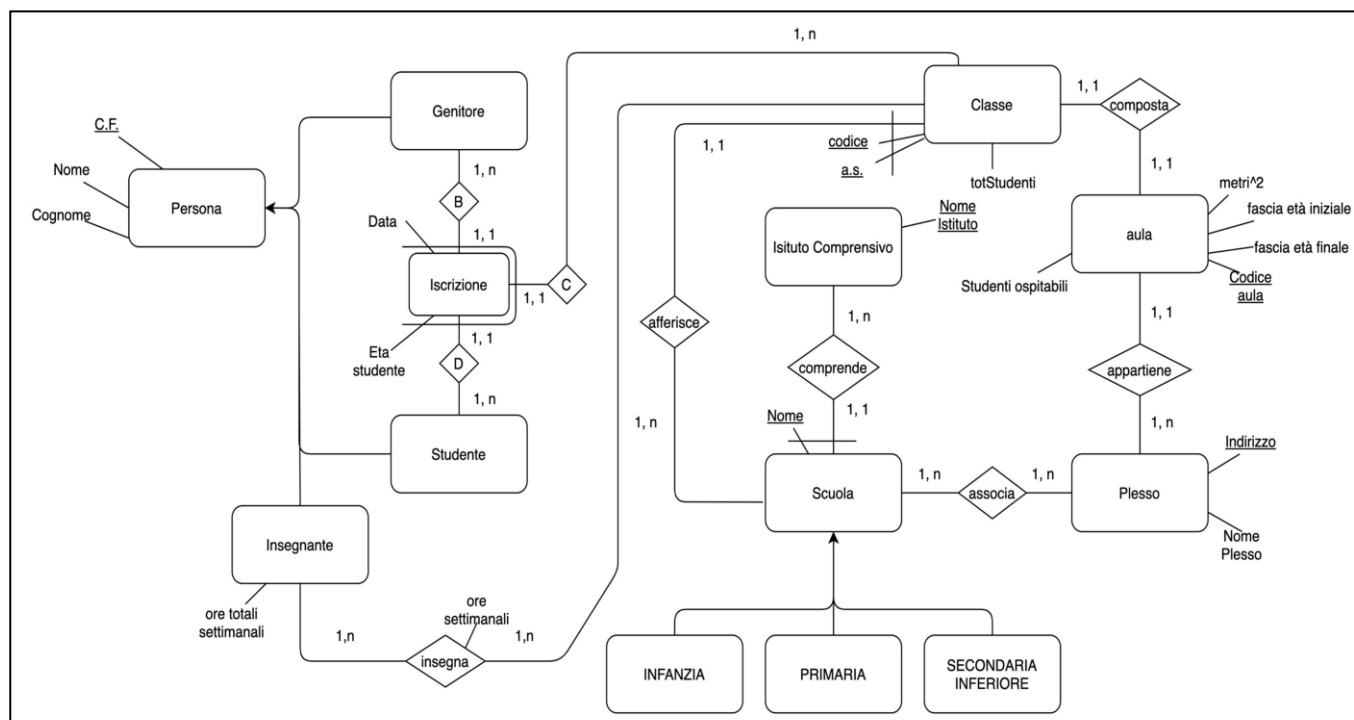
Aula	Di ogni aula si conoscono i metri quadri, hanno un numero massimo di studenti ospitabili e la fascia di età degli studenti ospitabili.
Classe	Ogni studente è associato ad una classe.
	Ogni classe, che ogni anno può cambiare aula, è identificata da un numero e da una lettera dell'alfabeto
Insegnante	Di ciascun insegnante si conosce, ogni anno, il numero di ore lavorative settimanali e le classi a cui è assegnato.

2.3 Specifica delle operazioni sui dati previste

1. Inserire una nuova iscrizione.
2. Trovare gli studenti iscritti alla scuola secondaria nell'anno 2020.
3. Trovare gli studenti che hanno frequentato almeno una classe per ogni grado dell'istituto.
4. Trovare i plessi con il numero di studenti ospitabili più alto.
5. Trovare, per ogni anno, le scuole con il maggior numero di iscritti.
6. Controllare che ad un'aula non vengano assegnati più iscritti di quanti ne possa ospitare.
7. Alla scadenza delle iscrizioni annuali (1° febbraio) si memorizzi il numero di nuovi iscritti per grado e classe (1-4 infanzia, 1-5 primaria e 1-3 medie).

3. Progettazione Concettuale

SCHEMI E-R



3.1 Documentazione dello Schema E-R

3.1.1 DIZIONARIO DEI DATI

ENTITÀ	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Istituto comprensivo	L'istituto comprensivo è un complesso scolastico al cui interno sono presenti scuole di più ordine e grado	NomeIstituto	NomeIstituto
Scuola	Istituzione avente un ordine ed un grado specifico organizzata per l'istruzione collettiva appartenente all'istituto comprensivo.	NomeScuola, NomeIstituto (IstitutoComprensivo)	NomeScuola, NomeIstituto (IstitutoComprensivo)
Infanzia	Organizzazione prescolastica destinata ai bambini dall'età di 3 anni all'età di 6 anni e che precede la scuola primaria.	NomeScuola, IstitutoComprensivo (Scuola)	NomeScuola, IstitutoComprensivo (Scuola)
Primaria	Scuola in cui gli alunni, di età compresa fra i 6 e gli 11 anni circa, ricevono l'educazione primaria o elementare.	NomeScuola, IstitutoComprensivo (Scuola)	NomeScuola, IstitutoComprensivo (Scuola)
Secondaria	Scuola rappresentante il secondo livello del primo ciclo di studio dell'istruzione obbligatoria. Dura tre anni, dagli 11 ai 14 anni.	NomeScuola, IstitutoComprensivo (Scuola)	NomeScuola, IstitutoComprensivo (Scuola)

Plesso	Costruzione edile al cui interno si svolgono le attività scolastiche e amministrative della scuola e dell'istituto comprensivo.	Indirizzo, nome Plesso	Indirizzo
Aula	Locale in cui si tengono le lezioni scolastiche.	Metri quadri, fascia d'età iniziale, fascia d'età finale, codice aula, studenti ospitabili.	Codice aula
Classe	Insieme di studenti frequentanti l'istituto comprensivo e associati ad una scuola.	codiceAula, a.s, nome (Scuola), nomeIstituto (Istituto Comprensivo)	codiceAula, a.s, nome (Scuola), nomeIstituto (IstitutoComprensivo)
Persona	Individuo della specie umana.	CodiceFiscale, Nome, Cognome.	CodiceFiscale
Studente	Persona frequentante la scuola	CodiceFiscale (Persona)	CodiceFiscale (Persona)
Genitore	Padre/madre di uno o più studenti.	CodiceFiscale (Persona)	CodiceFiscale (Persona)
Insegnante	Persona che si dedica all'insegnamento a livello professionale.	CodiceFiscale (Persona)	CodiceFiscale (Persona)
Iscrizione	Entità contenente lo storico delle iscrizioni degli studenti alle scuole dell'istituto comprensivo.	Data Iscrizione, età Studente, codice Fiscale (studente), codice Fiscale (genitore), codiceClasse (Classe), gradoScuola (Scuola), nomeScuola (Scuola), nomeIstituto (Istituto)	Data Iscrizione, età Studente, codice Fiscale (studente), codice Fiscale (genitore), codiceClasse (Classe), gradoScuola (Scuola), nomeScuola (Scuola), nomeIstituto (Istituto)

3.2 Descrizione Relazioni

RELAZIONE: INSEGNA		
DESCRIZIONE	RELAZIONE CHE ASSOCIA L'INSEGNANTE DI UNA DETERMINATA MATERIA AD UNA CLASSE IN UN DATO ANNO SCOLASTICO	
Entità Coinvolte		
ENTITÀ	CARDINALITÀ	
INSEGNANTE	(1,N)	
CLASSE	(1,N)	
ATTRIBUTI		
NOME	TIPO DI DATI	DESCRIZIONE
ORE SETTIMANALI	NUMERICO	NUMERO DI ORE SETTIMANALI CHE L'INSEGNANTE TRASCORRE IN UNA DETERMINATA CLASSE

RELAZIONE: APPARTIENE	
DESCRIZIONE	RELAZIONE CHE L'AULA AD UN PLESSO SCOLASTICO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
AULA	(1,1)
PLESSO	(1,N)

RELAZIONE: COMPOSTA	
DESCRIZIONE	RELAZIONE CHE ASSOCIA UN’AULA AD UNA CLASSE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
CLASSE	(1,1)
AULA	(1,1)

3.3 Vincoli non espressi dallo schema E/R

REGOLE DI VINCOLO
Verificare che al momento dell'assegnazione dell'aula ad una classe, gli studenti di quest'ultima rispettino la fascia d'età di tale aula.
Ogni anno uno studente può essere iscritto ad una singola classe.
Le ore totali di insegnamento di un insegnante non devono superare le ore complessive di insegnamento totali che l'insegnante ha con le proprie classi.
Le iscrizioni devono avvenire tra 1 e 31 gennaio dell'anno corrente.

Tabella 1. Vincoli non espressi dallo schema E-R

4. Progettazione Logica

L'obiettivo della fase di progettazione logica è quello di "tradurre" lo schema concettuale, prodotto in fase di progettazione concettuale, in uno schema logico che rappresenti gli stessi dati in maniera corretta ed efficiente.

La progettazione logica si articola in due fasi:

- Ristrutturazione del modello Entità-Relazione: è una fase indipendente dal modello logico scelto e si basa su criteri di ottimizzazione dello schema;
- Traduzione verso il modello logico: fa riferimento ad un particolare modello logico e può includere una ulteriore ottimizzazione.

Il modello logico utilizzato in questo progetto didattico è il Modello **Relazionale**.

4.1 Ristrutturazione dello schema ER

Si articola in quattro passi:

- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento di entità e relazioni
- Scelta degli identificatori primari

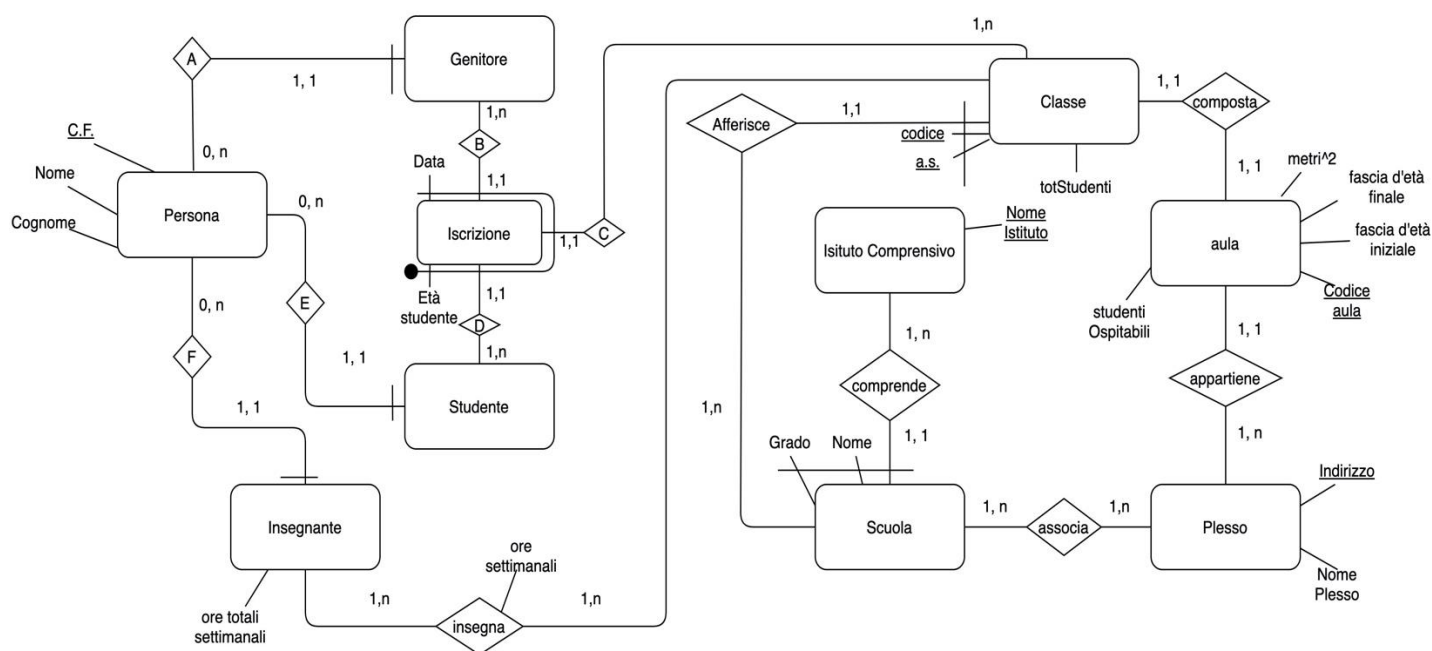


Figura 2- Schema E-R ristrutturato

4.2 Documentazione dello schema logico

4.2.1 DESCRIZIONE DELLE ENTITÀ

ISTITUTO COMPRENSIVO			
Descrizione	L'istituto comprensivo è un complesso scolastico al cui interno sono presenti scuole di più ordine e grado		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
NomeIstituto	VarChar [100]	Not null, Primary key	Nome dell'istituto

SCUOLA			
Descrizione	Istituzione avente un ordine ed un grado specifico organizzata per l'istruzione collettiva appartenente all'istituto comprensivo.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Nome	VarChar [20]	Not null, Primary key	Nome della scuola
Grado	VarChar [20]	Not null, primary key	Grado scolastico della scuola
Istituto	VarChar [100]	Not null, primary key	Nome dell'istituto comprensivo a cui appartiene la scuola

PLESSO			
Descrizione	Costruzione edile al cui interno si svolgono le attività scolastiche e amministrative della scuola e dell'istituto comprensivo.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Indirizzo	VarChar [50]	Not null, primary key	Indirizzo di locazione del plesso scolastico
NomePlesso	VarChar [50]	Not null	Nome del plesso scolastico

AULA			
Descrizione	Locale in cui si tengono le lezioni scolastiche.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Codice Aula	VarChar [10]	Not null, primary key	Codice identificativo dell'aula.
Indirizzo	VarChar [50]	Not null	Indirizzo del plesso.
Fascia d'età iniziale	Int	Not null, unsigned	Fascia d'età degli studenti che possono stare in una determinata aula
Metri quadrati	Double	Not null, unsigned	Unità di misura che indica la grandezza dell'aula.
Fascia d'età iniziale	Int	Not null, unsigned	Età minima degli studenti che possono stare in una determinata aula.
Fascia d'età finale	Int	Not null, unsigned	Età massima degli studenti che possono stare in una determinata aula.
Studenti ospitabili	Int	Not null, unsign.	Capacità massima dell'aula

4.3.1 Descrizione Relazioni

RELAZIONE: INSEGNA		
DESCRIZIONE	RELAZIONE CHE ASSOCIA L'INSEGNANTE DI UNA DETERMINATA MATERIA AD UNA CLASSE IN UN DATO ANNO SCOLASTICO	
Entità Coinvolte		
ENTITÀ	CARDINALITÀ	
INSEGNANTE	(1,N)	
CLASSE	(1,N)	
ATTRIBUTI		
NOME	TIPO DI DATI	DESCRIZIONE
ORE SETTIMANALI	INT	NUMERO DI ORE SETTIMANALI CHE L'INSEGNANTE TRASCORRE IN UNA DETERMINATA CLASSE

RELAZIONE: APPARTIENE	
DESCRIZIONE	RELAZIONE CHE ASSOCIA L’AULA AD UN PLESSO SCOLASTICO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
AULA	(1,1)
PLESSO	(1,N)

RELAZIONE: COMPOSTA	
DESCRIZIONE	RELAZIONE CHE ASSOCIA UN’AULA AD UNA CLASSE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
CLASSE	(1,1)
AULA	(1,1)

4.4.1 Traduzione entità

- Persona (codice Fiscale, Nome, Cognome);
- Genitore (codiceFiscale*);
Vincoli di integrità referenziale: codiceFiscale* di Genitore e codiceFiscale di Persona.
- Studente (codiceFiscale*);
Vincoli di integrità referenziale: codiceFiscale* di Genitore e codiceFiscale di Persona.
- Insegnante (codiceFiscale*, oreTotaliSettimanali);
Vincoli di integrità referenziale: codiceFiscale* di Genitore e codiceFiscale di Persona.
- Iscrizione (Data, EtàStudente, codiceFiscaleGenitore*, codiceFiscaleStudente*,
(codiceClasse, nomeScuola, nomeIstituto, aScolastico)*);
Vincoli di integrità referenziale:
 1. codiceFiscaleGenitore di Iscrizione e codiceFiscale di Genitore;
 2. codiceFiscaleStudente di Iscrizione e codiceFiscale di Studente;
 3. codiceClasse di Iscrizione e codice di Classe;
 4. nomeScuola di iscrizione e nomeScuola di Classe;
 5. nomeIstituto di Iscrizione e nomeIstituto di Classe;
 6. aScolastico di iscrizione e AnnoScolastico di Classe.
- Classe (Codice, AnnoScolastico, (nomeScuola, nomeIstituto, gradoScuola)*,
codiceAula*, totStudenti);
Vincoli di integrità referenziale:
 1. nomeScuola di Classe e nome di Scuola;
 2. gradoScuola di Classe e grado di Scuola;
 3. nomeIstituto di Classe e nome_istituto di Scuola;
 4. codiceAula di classe e codice_aula di Aula.
- Aula (codiceAula, indirizzo*, studentiOspitabili, metri^2, fascia d'età iniz., fascia d'età fin.);
Vincoli di integrità referenziale:
 1. Indirizzo di Aula e indirizzo di Plesso;
- IstitutoComprensivo (nomeIstituto);
- Plesso (indirizzo, nomePlesso);
- Scuola (nomeScuola, grado, nomeIstituto*);
Vincoli di integrità referenziale:
 1. nomeIstituto di Scuola e nome di Istituto

4.4.2 Traduzione relazioni

- Insegna (cfInsegnante*, IstitutoClasse*, (nomeScuola, gradoScuola, codiceClasse, annoS)*, oreSettimanali);

Vincoli di integrità referenziali:

1. cfInsegnante di Insegna e codiceFiscale di Insegnante;
2. IstitutoClasse di Insegna e nomeIstituto di IstitutoComprensivo;
3. nomeScuola di Insegna e nomeScuola di Classe;
4. gradoScuola di Insegna e gradoScuola di Classe;
5. codiceClasse di Insegna e codice di Classe;
6. annoS di Insegna e annoScolastico di Classe.

- Associa ((nomeScuola, gradoScuola, nomeIstituto)*, indirizzoPlesso*);

Vincoli di integrità referenziali:

1. nomeScuola di Associa e nome di Scuola;
2. gradoScuola di Associa e grado di Scuola;
3. nomeIstituto di Associa e istituto di Scuola;
4. indirizzoPlesso di Associa e indirizzo di Plesso.

5. Progettazione Fisica

5.1 Definizione dello schema della base di dati inserire create table, foreign key, trigger, scheduling

(Script SQL-DLL)

Tabella Associa (nomeScuola*, gradoScuola*, nomeIstituto*, indirizzoPlesso*);

```
CREATE TABLE `ASSOCIA` (  
  `NOME SCUOLA` VARCHAR(20) NOT NULL,  
  `GRADO SCUOLA` VARCHAR(20) NOT NULL,  
  `NOME ISTITUTO` VARCHAR(100) NOT NULL,  
  `INDIRIZZO PLESSO` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`NOME SCUOLA`, `GRADO SCUOLA`, `NOME ISTITUTO`, `INDIRIZZO PLESSO`),  
  KEY `SCUOLA.ISTITUTO_IDX` (`NOME ISTITUTO`),  
  KEY `SCUOLA.GRADO_IDX` (`GRADO SCUOLA`),  
  KEY `PLESSO.INDIRIZZO_IDX` (`INDIRIZZO PLESSO`),  
  CONSTRAINT `PLESSO.INDIRIZZO` FOREIGN KEY (`INDIRIZZO PLESSO`) REFERENCES `PLESSO` (`INDIRIZZO`),  
  CONSTRAINT `SCUOLA.GRADO` FOREIGN KEY (`GRADO SCUOLA`) REFERENCES `SCUOLA` (`GRADO`),  
  CONSTRAINT `SCUOLA.ISTITUTO` FOREIGN KEY (`NOME ISTITUTO`) REFERENCES `SCUOLA` (`ISTITUTO`),  
  CONSTRAINT `SCUOLA.NOME` FOREIGN KEY (`NOME SCUOLA`) REFERENCES `SCUOLA` (`NOME`)  
)
```

Tabella Aula (codiceAula, indirizzo*, studentiOspitabili, metri^2, fascia d'età iniz., fascia d'età fin.);

```
CREATE TABLE `AULA` (  
  `CODICE_AULA` VARCHAR(10) NOT NULL COMMENT 'CODICE IDENTIFICATIVO DELL"AULA."',  
  `INDIRIZZO` VARCHAR(50) NOT NULL,  
  `FASCIA_DI_ETÀ_INIZIALE` INT UNSIGNED NOT NULL COMMENT 'ETÀ MINIMA DEGLI STUDENTI CHE POSSONO STARE IN UNA DETERMINATA AULA.',  
  `FASCIA_DI_ETÀ_FINALE` INT UNSIGNED NOT NULL COMMENT 'ETÀ MASSIMA DEGLI STUDENTI CHE POSSONO STARE IN UNA DETERMINATA AULA.',  
  `METRI_QUADRATI` DOUBLE UNSIGNED NOT NULL COMMENT 'UNITÀ DI MISURA CHE INDICA LA GRANDEZZA DELL"AULA.',  
  `STUDENTI OSPITABILI` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`CODICE_AULA`),  
  KEY `AULA.INDIRIZZO_PLESSO_IDX` (`INDIRIZZO`),  
  CONSTRAINT `AULA.INDIRIZZO_PLESSO` FOREIGN KEY (`INDIRIZZO`) REFERENCES `PLESSO` (`INDIRIZZO`)  
)
```

Tabella Classe: (Codice, AnnoScolastico, nomeScuola*, nomeIstituto*, gradoScuola*, codiceAula*, totStudenti);

```
CREATE TABLE `CLASSE` (  
  `CODICE` VARCHAR(20) NOT NULL,  
  `NOME ISTITUTO` VARCHAR(100) NOT NULL,  
  `GRADO SCUOLA` VARCHAR(20) NOT NULL,  
  `NOME SCUOLA` VARCHAR(20) NOT NULL,  
  `ANNO SCOLASTICO` INT NOT NULL,  
  `AULA` VARCHAR(45) NOT NULL,  
  `TOT STUDENTI` INT NOT NULL,  
  PRIMARY KEY (`CODICE`, `NOME ISTITUTO`, `GRADO SCUOLA`, `NOME SCUOLA`, `ANNO SCOLASTICO`),  
  KEY `NOME SCUOLA` (`NOME SCUOLA`),  
  KEY `GRADO SCUOLA` (`GRADO SCUOLA`),  
  KEY `NOME ISTITUTO` (`NOME ISTITUTO`),  
  KEY `AULA` (`AULA`),
```

```

KEY `ANNO_SCOLASTICO` (`ANNO_SCOLASTICO`),
CONSTRAINT `CLASSE_IBFK_1` FOREIGN KEY (`NOME_SCUOLA`) REFERENCES `SCUOLA` (`NOME`),
CONSTRAINT `CLASSE_IBFK_2` FOREIGN KEY (`GRADO_SCUOLA`) REFERENCES `SCUOLA` (`GRADO`),
CONSTRAINT `CLASSE_IBFK_3` FOREIGN KEY (`NOME_ISTITUTO`) REFERENCES `ISTITUTO_COMPRENSIVO` (`NOME_ISTITUTO`),
CONSTRAINT `CLASSE_IBFK_4` FOREIGN KEY (`AULA`) REFERENCES `AULA` (`CODICE_AULA`)
)

```

Tabella Genitore: (codiceFiscale*);

```

CREATE TABLE `GENITORE` (
  `CODICEFISCALE` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`CODICEFISCALE`),
  CONSTRAINT `G.CODICEFISCALE` FOREIGN KEY (`CODICEFISCALE`) REFERENCES `PERSONA` (`CODICEFISCALE`)
)

```

Tabella Insegna: (codiceFiscale*, oreTotaliSettimanali);

```

CREATE TABLE `INSEGNA` (
  `CFINSEGNANTE` VARCHAR(20) NOT NULL,
  `ISTITUTOCLASSE` VARCHAR(100) NOT NULL,
  `NOME_SCUOLA` VARCHAR(20) NOT NULL,
  `GRADO_SCUOLA` VARCHAR(20) NOT NULL,
  `CODICECLASSE` VARCHAR(45) NOT NULL,
  `ORESETTIMANALI` INT NOT NULL,
  `ANNOS` INT NOT NULL,
  PRIMARY KEY (`CFINSEGNANTE`, `ISTITUTOCLASSE`, `NOME_SCUOLA`, `GRADO_SCUOLA`, `ANNOS`, `CODICECLASSE`),
  KEY `INSEGNA.ISTITUTO_IDX` (`ISTITUTOCLASSE`),
  KEY `INSEGNA.GRADO_IDX` (`GRADO_SCUOLA`),
  KEY `INSEGNA.CODICECLASSE_IDX` (`CODICECLASSE`),
  KEY `INSEGNA.CODICEFISCALE_IDX` (`CFINSEGNANTE`),
  KEY `INSEGNA.ANNO_SCOLASTICO_IDX` (`ANNOS`),
  KEY `INSEGNA.SCUOLA_IDX` (`NOME_SCUOLA`),
  CONSTRAINT `INSEGNA.ANNO` FOREIGN KEY (`ANNOS`) REFERENCES `CLASSE` (`ANNO_SCOLASTICO`),
  CONSTRAINT `INSEGNA.CODICECLASSE` FOREIGN KEY (`CODICECLASSE`) REFERENCES `CLASSE` (`CODICE`),
  CONSTRAINT `INSEGNA.CODICEFISCALE` FOREIGN KEY (`CFINSEGNANTE`) REFERENCES `INSEGNANTE` (`CODICEFISCALE`),
  CONSTRAINT `INSEGNA.GRADO` FOREIGN KEY (`GRADO_SCUOLA`) REFERENCES `CLASSE` (`GRADO_SCUOLA`),
  CONSTRAINT `INSEGNA.ISTITUTO` FOREIGN KEY (`ISTITUTOCLASSE`) REFERENCES `ISTITUTO_COMPRENSIVO` (`NOME_ISTITUTO`),
  CONSTRAINT `INSEGNA.SCUOLA` FOREIGN KEY (`NOME_SCUOLA`) REFERENCES `CLASSE` (`NOME_SCUOLA`)
) ENGINE=INNODB DEFAULT CHARSET=UTF8MB4 COLLATE=UTF8MB4_0900_AI_CI

```

Tabella Insegnante: (codiceFiscale*, oreTotaliSettimanali);

```

CREATE TABLE `INSEGNANTE` (
  `CODICEFISCALE` VARCHAR(20) NOT NULL,
  `ORETOTALISETTIMANALI` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`CODICEFISCALE`),
  CONSTRAINT `I.CODICEFISCALE` FOREIGN KEY (`CODICEFISCALE`) REFERENCES `PERSONA` (`CODICEFISCALE`)
)

```

Tabella Iscrizione: (Data,EtàStudente, codiceFiscaleGenitore*, codiceFiscaleStudente*, codiceClasse*, nomeScuola*, nomeIstituto*, aScolastico*);

```
CREATE TABLE `ISCRIZIONE` (
  `CODICEFISCALEGENITORE` VARCHAR(20) NOT NULL,
  `CODICEFISCALESTUDENTE` VARCHAR(20) NOT NULL,
  `CODICECLASSE` VARCHAR(3) NOT NULL,
  `GRADO SCUOLA` VARCHAR(20) NOT NULL,
  `NOME SCUOLA` VARCHAR(20) NOT NULL,
  `NOME ISTITUTO` VARCHAR(100) NOT NULL,
  `DATA ISCRIZIONE` DATETIME NOT NULL,
  `ETÀ STUDENTE` INT NOT NULL,
  `A SCOLASTICO` INT NOT NULL,
  PRIMARY KEY
  (`CODICEFISCALEGENITORE`,`CODICEFISCALESTUDENTE`,`CODICECLASSE`,`NOME SCUOLA`,`GRADO SCUOLA`,`NOME ISTITUTO`,`DATA ISCRIZIONE`,`ETÀ STUDENTE`,`A SCOLASTICO`),
  KEY `STUDENTE.CODICEFISCALE_IDX` (`CODICEFISCALESTUDENTE`),
  KEY `CLASSE.CODICE_IDX` (`CODICECLASSE`),
  KEY `CLASSE.SCUOLA_IDX` (`NOME SCUOLA`),
  KEY `CLASSE.AS_IDX` (`A SCOLASTICO`),
  KEY `CLASSE.GRADO_IDX` (`GRADO SCUOLA`),
  KEY `CLASSE.ISTITUTO_IDX` (`NOME ISTITUTO`),
  CONSTRAINT `CLASSE.A SCOLASTICO` FOREIGN KEY (`A SCOLASTICO`) REFERENCES `CLASSE` (`ANNO SCOLASTICO`),
  CONSTRAINT `CLASSE.CODICE` FOREIGN KEY (`CODICECLASSE`) REFERENCES `CLASSE` (`CODICE`),
  CONSTRAINT `CLASSE.GRADO` FOREIGN KEY (`GRADO SCUOLA`) REFERENCES `CLASSE` (`GRADO SCUOLA`),
  CONSTRAINT `CLASSE.ISTITUTO` FOREIGN KEY (`NOME ISTITUTO`) REFERENCES `CLASSE` (`NOME ISTITUTO`),
  CONSTRAINT `CLASSE.SCUOLA` FOREIGN KEY (`NOME SCUOLA`) REFERENCES `CLASSE` (`NOME SCUOLA`),
  CONSTRAINT `GENITORE.CODICEFISCALE` FOREIGN KEY (`CODICEFISCALEGENITORE`) REFERENCES `GENITORE` (`CODICEFISCALE`),
  CONSTRAINT `STUDENTE.CODICEFISCALE` FOREIGN KEY (`CODICEFISCALESTUDENTE`) REFERENCES `STUDENTE` (`CODICEFISCALE`)
) ENGINE=INNODB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='TABELLA RAPPRESENTANTE L'ISCRIZIONE CHE UN GENITORE EFFETTUA DEL PROPRIO FIGLIO AD UNA CLASSE'
```

Tabella Istituto_Comprensivo: (nomeIstituto);

```
CREATE TABLE `ISTITUTO_COMPRENSIVO` (
  `NOME ISTITUTO` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`NOME ISTITUTO`)
) ENGINE=INNODB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='UN COMPLESSO SCOLASTICO AL CUI INTERNO SONO PRESENTI SCUOLE DI PIÙ ORDINI E GRADI.'
```

Tabella Persona: (codice Fiscale,Nome,Cognome);

```
CREATE TABLE `PERSONA` (
  `CODICEFISCALE` VARCHAR(20) NOT NULL COMMENT 'CODICE UNIVOCO PERSONALE IDENTIFICATIVO.',
  `NOME` VARCHAR(50) NOT NULL COMMENT 'NOME DELLA PERSONA.',
  `COGNOME` VARCHAR(50) NOT NULL COMMENT 'COGNOME DELLA PERSONA.',
  PRIMARY KEY (`CODICEFISCALE`)
)
```

Tabella Plesso: (indirizzo, nomePlesso);

```
CREATE TABLE `PLESSO` (
  `INDIRIZZO` VARCHAR(50) NOT NULL COMMENT 'INDIRIZZO DI LOCAZIONE DEL PLESSO SCOLASTICO',
  `NOME_PLESSO` VARCHAR(50) NOT NULL COMMENT 'NOME DEL PLESSO SCOLASTICO',
  PRIMARY KEY (`INDIRIZZO`)
) ENGINE=INNODB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='COSTRUZIONE EDILE AL CUI INTERNO SI SVOLGONO LE ATTIVITÀ SCOLASTICHE E AMMINISTRATIVE DELLA SCUOLA E DELL'ISTITUTO COMPRENSIVO.'
```

Tabella ResocontoAnnuale: (classe,grado,anno,data, numIscritti)
<pre>CREATE TABLE `RESOCONTOANNUALE` (`CLASSE` VARCHAR(3) NOT NULL, `GRADO` VARCHAR(20) NOT NULL, `ANNO` INT NOT NULL, `DATA` DATE NOT NULL, `NUMEROISCRITTI` INT NOT NULL, PRIMARY KEY (`ANNO`,`GRADO`,`CLASSE`))</pre>

Tabella Scuola: (nomeScuola,grado,nomeIstituto*);
<pre>CREATE TABLE `SCUOLA` (`NOME` VARCHAR(20) NOT NULL COMMENT 'NOME DELLA SCUOLA', `GRADO` VARCHAR(20) NOT NULL COMMENT 'GRADO SCOLASTIC DELLA SCUOLA.', `ISTITUTO` VARCHAR(100) NOT NULL, PRIMARY KEY (`GRADO`,`NOME`,`ISTITUTO`), UNIQUE KEY `GRADO_UNIQUE` (`GRADO`), KEY `NOME_ISTITUTO_IDX` (`NOME`), KEY `ISTITUTO_IDX` (`ISTITUTO`), CONSTRAINT `ISTITUTO` FOREIGN KEY (`ISTITUTO`) REFERENCES `ISTITUTO_COMPRENSIVO` (`NOME_ISTITUTO`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='ISTITUZIONE AVENTE UN ORDINE ED UN GRADO SPECIFICO ORGANIZZATA PER L''ISTRUZIONE COLLETTIVA APPARTENENTE ALL'ISTITUTO COMPRENSIVO.'</pre>

Tabella Studente: (codiceFiscale*);
<pre>CREATE TABLE `STUDENTE` (`CODICEFISCALE` VARCHAR(20) NOT NULL, PRIMARY KEY (`CODICEFISCALE`), CONSTRAINT `S.CODICEFISCALE` FOREIGN KEY (`CODICEFISCALE`) REFERENCES `PERSONA` (`CODICEFISCALE`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci</pre>

Trigger BeforeInsert di Iscrizione

<pre>CREATE DEFINER=`root`@`localhost` TRIGGER `ISCRIZIONE_BEFORE_INSERT` BEFORE INSERT ON `ISCRIZIONE` FOR EACH ROW BEGIN /* TRIGGER CHE EVITA LE ISCRIZIONI IN MESE DIVERSO DA GENNAIO */ DECLARE MESSAGE_TEXT VARCHAR(200); IF (EXTRACT(MONTH FROM NEW.DATAISCRIZIONE) <>01) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'MESE NON VALIDO'; END IF; /* TRIGGER CHE EVITA LE ISCRIZIONI DUPLICATE NELLO STESSO ANNO */ IF (EXISTS (SELECT * FROM ISCRIZIONE AS I WHERE I.CODICEFISCALESTUDENTE=NEW.CODICEFISCALESTUDENTE AND EXTRACT(YEAR FROM I.DATAISCRIZIONE)=EXTRACT(YEAR FROM NEW.DATAISCRIZIONE))) THEN SIGNAL SQLSTATE '40000' SET MESSAGE_TEXT = 'ISCRIZIONE GIÀ PRESENTE'; END IF;</pre>

```

/* TRIGGER CHE EVITA LE ISCRIZIONI ERRATE PER CLASSE */
IF (NOT EXISTS (SELECT *
                FROM CLASSE AS C
                WHERE C.CODICE=NEW.CODICECLASSE AND EXTRACT(YEAR FROM NEW.DATAISCRIZIONE)=C.ANNOSCOLASTICO))
    THEN SIGNAL SQLSTATE '41000' SET MESSAGE_TEXT = 'CLASSE NON PRESENTE';
END IF;

/* TRIGGER CHE EVITA LE ISCRIZIONI ERRATE PER ANNO */
IF (YEAR(NEW.DATAISCRIZIONE)<>NEW.ASCOLASTICO)
    THEN SIGNAL SQLSTATE '41000' SET MESSAGE_TEXT = 'ANNO SCOLASTICO NON VALIDO';
END IF;

/* TRIGGER CHE VERIFICHI CHE LO STUDENTE RISPETTI LA FASCIA D'ETÀ NELLA CLASSE */
IF (EXISTS (SELECT *
            FROM AULA, CLASSE
            WHERE CLASSE.AULA = AULA.CODICE_AULA AND NEW.CODICECLASSE=CLASSE.CODICE AND (AULA.FASCIA_DI_ETÀ_INIZIALE >
NEW.ETÀSTUDENTE OR AULA.FASCIA_DI_ETÀ_FINALE < NEW.ETÀSTUDENTE) ))
    THEN SIGNAL SQLSTATE '50000' SET MESSAGE_TEXT = 'FASCIA DI ETÀ ERRATA PER LA CLASSE';
END IF;

/* TRIGGER CHE LIMITA L'ISCRIZIONE DEGLI STUDENTI NELL'AULA (CAPIENZA MASSIMA) */
IF (EXISTS (SELECT *
            FROM CLASSE,AULA
            WHERE NEW.CODICECLASSE=CLASSE.CODICE AND CLASSE.AULA=AULA.CODICE_AULA AND
CLASSE.TOTSTUDENTI>=AULA.STUDENTIOSPITABILI))
    THEN SIGNAL SQLSTATE '60000' SET MESSAGE_TEXT = 'CAPIENZA MASSIMA AULA SUPERATA';
    ELSE
        UPDATE ISTITUTO.CLASSE SET TOTSTUDENTI = TOTSTUDENTI+1 WHERE (CODICE = NEW.CODICECLASSE AND
CLASSE.ANNO SCOLASTICO=EXTRACT(YEAR FROM NEW.DATAISCRIZIONE));
    END IF;

END

```

Trigger AfterDelete di Iscrizione

```

CREATE DEFINER='root'@'localhost' TRIGGER `ISCRIZIONE_AFTER_DELETE` AFTER DELETE ON `ISCRIZIONE` FOR EACH ROW
BEGIN
    UPDATE ISTITUTO.CLASSE SET CLASSE.TOTSTUDENTI = CLASSE.TOTSTUDENTI-1 WHERE (CLASSE.CODICE = OLD.CODICECLASSE);
END

```

Trigger BeforeInsert di Insegna

```

CREATE DEFINER='root'@'localhost' TRIGGER `INSEGNA_BEFORE_INSERT` BEFORE INSERT ON `INSEGNA` FOR EACH ROW
BEGIN

DECLARE MESSAGE_TEXT VARCHAR(200);
DECLARE SOMMA INT;
DECLARE ORETOTALI INT;
SET SOMMA = (SELECT SUM(ORESETTIMANALI) FROM INSEGNA WHERE CFINSEGNANTE=NEW.CFINSEGNANTE);
SET ORETOTALI = (SELECT ORETOTALISETTIMANALI FROM INSEGNANTE WHERE CODICEFISCALE = NEW.CFINSEGNANTE);
IF (SOMMA + NEW.ORESETTIMANALI > ORETOTALI) THEN
    SIGNAL SQLSTATE '44000' SET MESSAGE_TEXT = 'ORE TOTALI SUPERATE';
END IF;

END

```

Trigger BeforeInsert di Classe

```
CREATE DEFINER='root'@'localhost' TRIGGER `CLASSE_BEFORE_INSERT` BEFORE INSERT ON `CLASSE` FOR EACH ROW BEGIN
    IF (EXISTS (SELECT AULA
        FROM CLASSE
        WHERE NEW.AULA=AULA AND NEW.ANNO SCOLASTICO=CLASSE.ANNO SCOLASTICO))
        THEN SIGNAL SQLSTATE '15000' SET MESSAGE_TEXT = 'AULA GIÀ ASSEGNATA';
    END IF;
END
```

Trigger BeforeInsert di Scuola

```
CREATE DEFINER='root'@'localhost' TRIGGER `SCUOLA_BEFORE_INSERT` BEFORE INSERT ON `SCUOLA` FOR EACH ROW BEGIN
    IF (EXISTS (SELECT NOME
        FROM SCUOLA AS S
        WHERE S.NOME=NEW.NOME ))
        THEN SIGNAL SQLSTATE '10000' SET MESSAGE_TEXT = 'SCUOLA GIÀ PRESENTE';
    END IF;
END
```

StoredProcedure resoconto annuale delle Iscrizioni

```
CREATE DEFINER='root'@'localhost' PROCEDURE `RESOCONTOISCRITTI`()
BEGIN
    DECLARE CLASSE VARCHAR(3);
    DECLARE GRADO VARCHAR (20);
    DECLARE ANNO INT;
    DECLARE NUMISCRITTI INT;

    DECLARE CURSORE CURSOR FOR SELECT COUNT(CODICE FISCALE STUDENTE), CODICE CLASSE, GRADO SCUOLA, EXTRACT(YEAR FROM
DATAISCRIZIONE)
        FROM ISCRIZIONE
        WHERE EXTRACT(YEAR FROM DATAISCRIZIONE)=EXTRACT(YEAR FROM CURDATE())
        GROUP BY CODICE CLASSE, GRADO SCUOLA, EXTRACT(YEAR FROM DATAISCRIZIONE);
    OPEN CURSORE;

    LOOP
    FETCH CURSORE INTO NUMISCRITTI, CLASSE, GRADO, ANNO;
    INSERT INTO RESOCONTOANNUALE VALUES (CLASSE, GRADO, ANNO, CURDATE(), NUMISCRITTI);
    END LOOP;
    CLOSE CURSORE;
END
```

Evento resoconto iscrizioni

```
SET GLOBAL EVENT_SCHEDULER = ON;

DELIMITER //

CREATE EVENT `EVENTOISCRIZIONI` ON SCHEDULE EVERY 1 YEAR
STARTS '2020-02-01 00:00:00'

DO BEGIN
    CALL RESOCONTOISCRITTI();
END
```

```
// DELIMITER ;
```

5.2 Definizione delle interrogazioni per la visualizzazione dei dati

5.2.1 Script SQL-DML : analisi puntuali – riferimento ad un tupla

ES. TUTTI I PRODOTTI DI TIPO PROFUMERIE,

1. INSERIRE UNA NUOVA ISCRIZIONE

```
Insert into iscrizione (codiceFiscaleGenitore, codiceFiscaleStudente, codiceClasse, gradoScuola,
nomeScuola, nomeIstituto, dataIscrizione, etàStudente, aScolastico)
Values ('G001', 'S005', '1-C', 'secondaria', 'Scuola2', 'Unical', '2020-01-15', 11, 2020);
```

2. TROVARE GLI STUDENTI ISCRITTI ALLA SCUOLA SECONDARIA NELL'ANNO 2020

```
Select codiceFiscaleStudente
From iscrizione
Where gradoScuola='secondaria' and extract(year from dataIscrizione)=2020;
```

3. TROVARE GLI STUDENTI CHE HANNO FREQUENTATO ALMENO UNA CLASSE PER OGNI GRADO DELL'ISTITUTO

```
Select codiceFiscaleStudente
From iscrizione as i1
Where EXISTS (Select *
              From iscrizione as i2, iscrizione as i3
              Where (i1.gradoScuola='infanzia' and i1.codiceFiscaleStudente=i2.codiceFiscaleStudente
and i2.gradoScuola='primaria' and i1.codiceFiscaleStudente=i3.codiceFiscaleStudente and
i3.gradoScuola='secondaria' ));
```

5.2.2 Script SQL-DML : analisi aggregate – riferimento a più tuple

Incasso totale di un semestre con uso di funzioni di aggregazione; group by

4. TROVARE I PLESSI CON IL NUMERO DI STUDENTI OSPITABILI PIÙ ALTO

```
Create View InfoPlesso (indirizzoPlesso, nomePlesso, numStudenti ) AS
Select plesso.indirizzo, plesso.nome_plesso, sum(aula.studentiOspitabili)
From aula, plesso
```

```

Where aula.indirizzo=plesso.indirizzo
Group by plesso.indirizzo, plesso.nome_plesso;

Select indirizzoPlesso
From InfoPlesso
Where numStudenti =( Select MAX(numStudenti)
                      From InfoPlesso);

```

5. TROVARE PER OGNI ANNO LE SCUOLE CON IL MAGGIOR NUMERO DI ISCRITTI

```

Create or replace view InfoScuola (nomeScuola, annoScolastico, numStudenti ) AS
Select scuola.nome,classe.annoScolastico, sum(classe.totStudenti)
From scuola, classe
Where classe.nomeScuola=scuola.nome
Group by nomeScuola, annoScolastico;

Select nomeScuola, annoScolastico, numStudenti
From InfoScuola as i1
Where NOT EXISTS (Select *
                  From InfoScuola as i2
                  Where i1.numStudenti<i2.numStudenti and i1.annoScolastico=i2.annoScolastico);

```

6. CONTROLLARE CHE AD UN AULA NON VENGANO ASSEGNATI PIÙ ISCRITTI DI QUANTI NE POSSA OSPITARE FATTO NEI TRIGGER

```

if (exists (Select *
            From classe,aula
            Where new.codiceClasse=classe.codice and classe.aula=aula.codice_aula and
            classe.totStudenti>=aula.studentiOspitabili))
    then SIGNAL SQLSTATE '60000' SET MESSAGE_TEXT = 'capienza massima aula superata';
else
    UPDATE Istituto.classe SET totStudenti = totStudenti+1 WHERE (codice = new.codiceClasse and
    classe.annoScolastico=extract(year from new.dataIscrizione));
end if;

```

7. ALLA SCADENZA DELLE ISCRIZIONI ANNUALI (1° FEBBRAIO) SI MEMORIZZI IL NUMERO DI NUOVI ISCRITTI PER GRADO E CLASSE (1-4 INFANZIA, 1-5 PRIMARIA E 1-3 MEDIE).

```

BEGIN
Declare classe varchar(3);
Declare grado varchar (20);
Declare anno int;
Declare numIscritti int;

DECLARE cursore CURSOR FOR SELECT
count(codiceFiscaleStudente),codiceClasse,gradoScuola,extract(Year from dataIscrizione)

```

```
From iscrizione
Where extract(YEAR from dataIscrizione)=extract(Year from curdate())
Group by codiceClasse, gradoScuola, extract(Year From dataIscrizione);
open cursore;
Loop
Fetch cursore INTO numIscritti,classe,grado,anno;
INSERT INTO resocontoannuale VALUES (classe,grado,anno,curdate(),numIscritti);
END LOOP;
Close cursore;
END
```