

# Specifiche di Progetto – Orale di Basi di Dati

## 1 Informazioni generali

- *Il progetto consiste nell'implementare una realtà (detta mini-mondo) di propria invenzione utilizzando un DBMS a propria scelta*
  - *The top 5 open source systems, April 2024* (fonte: [https://db-engines.com/en/ranking\\_osvsc](https://db-engines.com/en/ranking_osvsc))

Rank	System	Overall Rank	Type
1.	<a href="#">MySQL</a>	2.	RDBMS
2.	<a href="#">PostgreSQL</a>	4.	RDBMS
3.	<a href="#">MongoDB</a>	5.	NoSQL
4.	<a href="#">Redis</a>	7.	NoSQL
5.	<a href="#">Elasticsearch</a>	8.	NoSQL

- Il progetto può essere realizzato dal singolo o da un gruppo di massimo due persone: si richiede un minimo di 4 entità forti per il singolo studente, un minimo di 7 per la coppia. Nel caso di presenza di superclasse/sottoclassi, se una sottoclasse è coinvolta tramite associazioni con altre entità, viene contata come entità forte oltre alla superclasse; in caso contrario solo la superclasse è contata come entità forte.
- È cura dello studente illustrare il progetto *funzionante sul proprio PC*.

## 2 Tipi di progetto

### 2.1 Per chi è iscritto solo al corso di Basi di Dati

Il progetto dovrà produrre **sia una Relazione** (documento pdf, vedi Sez. 3) **sia la base di dati** implementata in un DBMS (MySQL o PostgreSQL).

La relazione sarà corretta dal docente **prima** della presentazione orale del progetto.

## 2.2 Per chi segue (o ha seguito) il curriculum di *Ingegneria Informatica - Sistemi Web*

- È possibile consegnare un progetto congiunto che riguarda i corsi di Basi di Dati + Ingegneria dei Sistemi Web

*Come procedere*

- 1) È possibile (1) scegliere tra i progetti suggeriti all'interno del corso di *Ingegneria dei Sistemi Web*, o (2) inventare un progetto di complessità analoga, per il quale è necessario ricevere l'approvazione di tutti i docenti inviando loro una mail.

**Per il progetto congiunto presentare un'analisi dei requisiti che tenga conto sia del formato richiesto dal corso di Ing. dei Sistemi web che del template richiesto per il corso di Basi di dati.**

- 2) Ottenuta la conferma, è possibile iniziare a lavorare al progetto che, per quanto riguarda il corso di Basi di Dati, dovrà produrre sia una **Relazione** (documento pdf, vedi Sez. 3) sia la **base di dati** implementata in un DBMS (MySQL o PostgreSQL).

Per la base di dati realizzare l'implementazione nel DBMS fino alla 2NF, ma normalizzare fino alla 3NF nella Relazione.

- 3) **Il progetto per l'insegnamento di Basi di dati può essere consegnato indipendentemente dal fatto che sia terminato il progetto relativo all'applicativo web.** La presentazione del progetto prevedrà orali distinti per i vari corsi.

La relazione sarà corretta dal docente **prima** della presentazione orale del progetto.

---

In entrambi i casi lo sviluppo del progetto in PostgreSQL garantisce +1 punto all'orale.

### 3 Template per la Relazione di progetto

#### Capitoli

- 1) Analisi dei requisiti (“descrizione mini-mondo”)
- 2) Progettazione dello Schema ER/EER
- 3) Schema Relazionale
- 4) Creazione del database
- 5) Interrogazioni SQL

#### 1) Analisi dei requisiti (“descrizione mini-mondo”) (1 pagina)

- Si scelga e si descriva un ambiente (una palestra, un e-commerce, ...) che evidenzia una richiesta di gestione tramite base di dati. **Descrivere chiaramente entità, attributi (e tra essi quelli univoci), rapporti di cardinalità e vincoli di partecipazione in modo da giustificare le scelte nel diagramma EER del punto 2.**

*Un esempio è l'analisi dei requisiti del database AZIENDA (vedi slide “Modello ER.pdf”).*

- L'incompletezza in termini di attributi, attributi univoci, rapporti di cardinalità o vincoli di partecipazione nell'analisi dei requisiti sarà tenuta in conto nella valutazione della relazione.

#### 2) Progettazione dello Schema ER/EER (1 pagina)

- Rappresentazione dello schema EER **finale, con rapporti di cardinalità e vincoli di partecipazione.**
- Utilizzare un software di grafica (Power Point, ecc) che consenta di rappresentare tutte le notazioni diagrammatiche come viste a lezione. Eventuali differenze non corrette saranno penalizzate.

#### 3) Schema Relazionale (1 pagina)

- 1) Schema relazionale **finale con vincoli di integrità referenziale** (tabelle non normalizzate).
- 2) Normalizzazione fino alla 3NF obbligatoria (ogni normalizzazione deve essere giustificata); *facoltativa l'applicazione delle tecniche 2NF e 3NF generali e la BCNF* (se corretta se ne terrà conto nel punteggio).
- 3) Schema relazionale **normalizzato**.

#### 4) Creazione del database

- Dump SQL del database (copiare il contenuto del file *.sql*).

#### 5) Interrogazioni SQL (1 pagina)

- Realizzare **4** interrogazioni in linguaggio SQL significative e **complesse** per la base di dati. La complessità e il corretto funzionamento saranno i parametri principali di valutazione. Non saranno valutate eventuali ulteriori interrogazioni.

## 4 Esempio di Relazione di progetto ispirato al database AZIENDA

### 1) Analisi dei requisiti

*Si vuole realizzare un sistema di gestione tramite base di dati che tenga traccia degli impiegati, dei dipartimenti e dei progetti di un'azienda.*

*L'azienda è organizzata in dipartimenti, ciascuno dei quali ha un nome e un numero univoci, più sedi e un particolare impiegato che lo gestisce. Si registrerà la data in cui quell'impiegato ha cominciato a gestire il dipartimento. In un dipartimento lavorano più impiegati.*

*Un dipartimento controlla 0 o più progetti, ciascuno dei quali ha un nome e un numero univoci e una singola sede. Un progetto può essere controllato da un solo dipartimento.*

*Per ciascun impiegato verrà memorizzato il nome, il numero di previdenza sociale (SSN), l'indirizzo, lo stipendio, il sesso e la data di nascita. Un impiegato è assegnato a un solo dipartimento ma può lavorare su molti progetti, non necessariamente controllati dallo stesso dipartimento.*

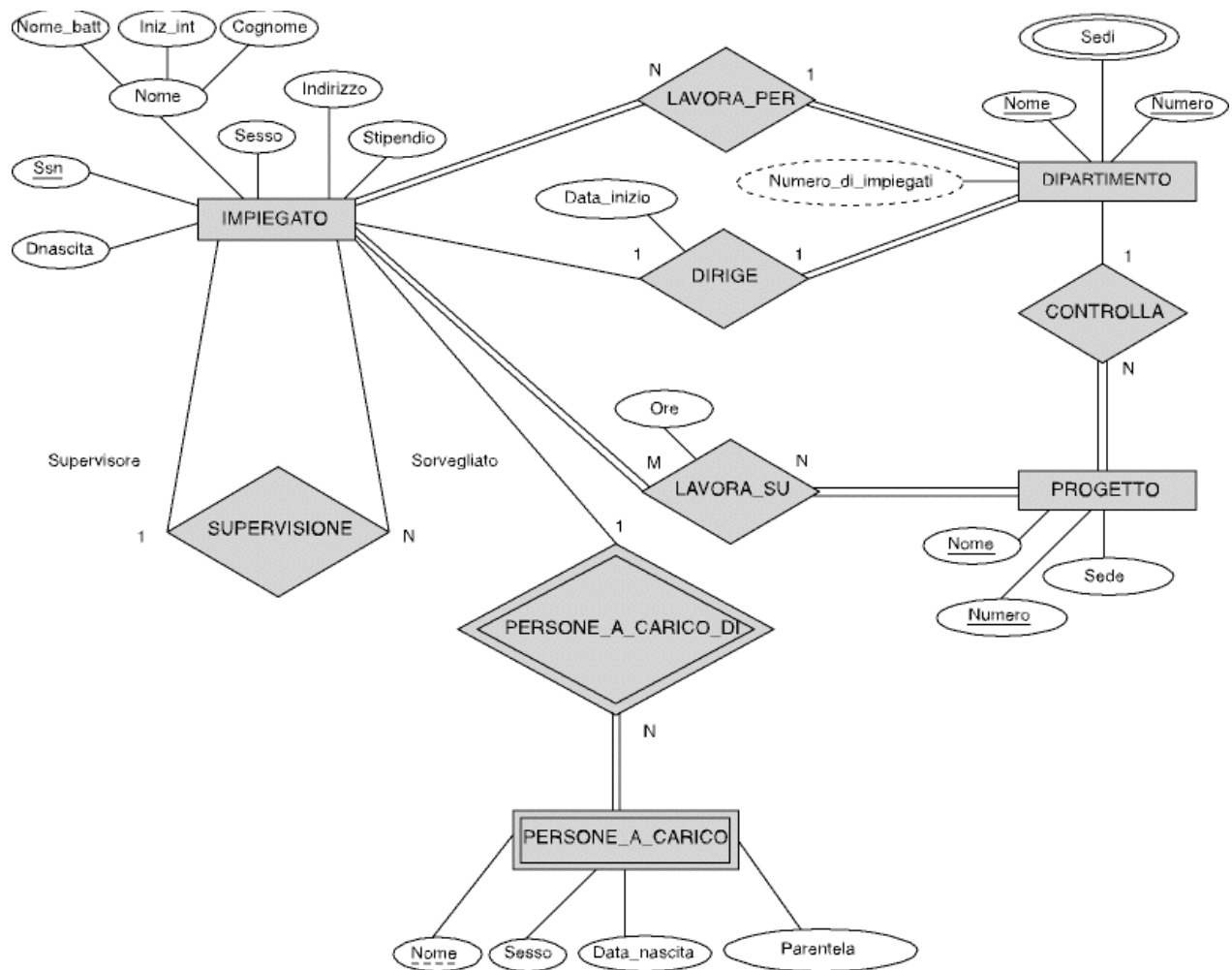
*Si terrà traccia del numero di ore settimanali lavorate da un impiegato su ciascun progetto e del diretto supervisore di ciascun impiegato (che è un altro impiegato). Un impiegato può lavorare contemporaneamente a più progetti e a ciascun progetto possono lavorare più impiegati.*

*Si registreranno a scopi assicurativi anche le persone a carico, se esistenti, di ciascun impiegato e per ciascuna di loro verranno memorizzati il nome di battesimo, il sesso, la data di nascita e il rapporto di parentela con l'impiegato.*

*Il diagramma ER comprenderà 3 entità forti (impiegati, dipartimenti, progetti).*

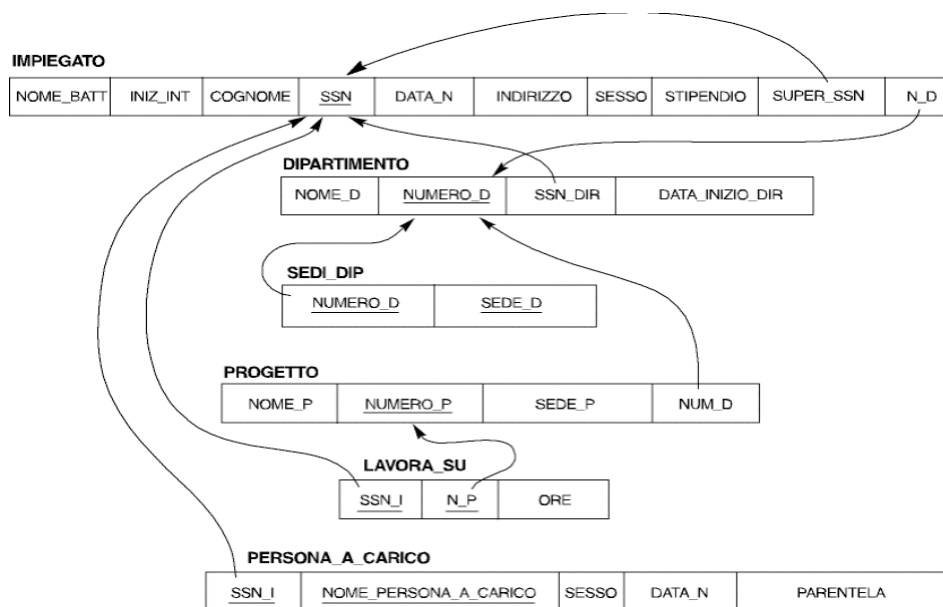
**IMPORTANTE!** La frase di chiusura della descrizione deve specificare il numero di entità forti.

## 2) Progettazione dello Schema ER/EER



### 3) Schema Relazionale

#### 1) Schema relazionale finale con vincoli di integrità referenziale (Ok anche nel formato R1.FK → R2. PK)



#### 2) Normalizzazione

1NF: lo schema è in 1NF in quanto non sussistono attributi multivalore, composti o loro combinazione.  
 2NF: Per le tabelle con una chiave primaria composta da un solo attributo il test è inutile (IMPIEGATO, DIPARTIMENTO, SEDI\_DIP, PROGETTO). Per LAVORA\_SU l'attributo non-primario Ore dipende funzionalmente in modo completo dalla chiave primaria. Per PERSONA\_A\_CARICO sussiste la DF Nome\_persona\_a\_carico → {Sesso, Data\_N}, per cui si normalizza la relazione in:

PERSONA\_A\_CARICO1

<u>NOME_PERSONA_A_CARICO</u>	SESSO	DATA_N
------------------------------	-------	--------

PERSONA\_A\_CARICO2

<u>SSN_I</u>	<u>NOME_PERSONA_A_CARICO</u>	PARENTELA
--------------	------------------------------	-----------

3NF: Tutte le relazioni sono in 3NF in quanto non sussistono DF transitive.

#### 3) Schema relazionale normalizzato

**IMPIEGATO**

NOME_BATT	INIZ_INT	COGNOME	<u>SSN</u>	DATA_N	INDIRIZZO	SESSO	STIPENDIO	SUPER_SSN	N_D
-----------	----------	---------	------------	--------	-----------	-------	-----------	-----------	-----

**DIPARTIMENTO**

NOME_D	<u>NUMERO_D</u>	SSN_DIR	DATA_INIZIO_DIR
--------	-----------------	---------	-----------------

**SEDI\_DIP**

<u>NUMERO_D</u>	<u>SEDE_D</u>
-----------------	---------------

**PROGETTO**

NOME_P	<u>NUMERO_P</u>	SEDE_P	NUM_D
--------	-----------------	--------	-------

**LAVORA\_SU**

<u>SSN_I</u>	<u>N_P</u>	ORE
--------------	------------	-----

**PERSONA\_A\_CARICO1**

<u>NOME_PERSONA_A_CARICO</u>	SESSO	DATA_N
------------------------------	-------	--------

**PERSONA\_A\_CARICO2**

<u>SSN_I</u>	<u>NOME_PERSONA_A_CARICO</u>	PARENTELA
--------------	------------------------------	-----------

#### 4) Creazione del database

**NB: l'implementazione del database può essere realizzata in seconda o in terza forma normale.**

```
-- phpMyAdmin SQL Dump
-- version 4.7.0
-- https://www.phpmyadmin.net/
--
-- Host: localhost
-- Creato il: Apr 05, 2019 alle 10:01
-- Versione del server: 5.7.25
-- Versione PHP: 7.1.23

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `azienda`
--

--
-- Struttura della tabella `DIPARTIMENTO`
--

CREATE TABLE `DIPARTIMENTO` (
  `NOME_D` varchar(15) NOT NULL,
  `NUMERO_D` int(11) NOT NULL,
  `SSN_DIR` char(9) NOT NULL,
  `DATA_INIZIO_DIR` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```

```

-- Dump dei dati per la tabella `DIPARTIMENTO`
--

INSERT INTO `DIPARTIMENTO` (`NOME_D`, `NUMERO_D`, `SSN_DIR`, `DATA_INIZIO_DIR`)
VALUES
('Sede centrale', 1, '888665555', '1981-06-19'),
('Amministrazione', 4, '987654321', '1995-01-01'),
('Ricerca', 5, '333445555', '1988-05-22');

-----

--

-- Struttura della tabella `IMPIEGATO`
--

CREATE TABLE `IMPIEGATO` (
  `NOME_BATT` varchar(15) NOT NULL,
  `INIZ_INT` char(1) DEFAULT NULL,
  `COGNOME` varchar(15) NOT NULL,
  `SSN` char(9) NOT NULL,
  `DATA_N` date DEFAULT NULL,
  `INDIRIZZO` varchar(30) DEFAULT NULL,
  `SESSO` char(1) DEFAULT NULL,
  `STIPENDIO` decimal(10,0) DEFAULT NULL,
  `SUPER_SSN` char(9) DEFAULT NULL,
  `N_D` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dump dei dati per la tabella `IMPIEGATO`
--

INSERT INTO `IMPIEGATO` (`NOME_BATT`, `INIZ_INT`, `COGNOME`, `SSN`, `DATA_N`,
`INDIRIZZO`, `SESSO`, `STIPENDIO`, `SUPER_SSN`, `N_D`) VALUES
('John', 'B', 'Smith', '123456789', '1965-01-09', '731 Fondren, Houston, TX',
'M', '30000', '333445555', 5),
('Franklin', 'T', 'Wong', '333445555', '1955-12-08', '638 Voss, Houston, TX',
'M', '40000', '888665555', 5),
('Joyce', 'A', 'English', '453453453', '1972-07-31', '5631 Rice, Houston, TX',
'F', '25000', '333445555', 5),
('Ramesh', 'K', 'Narayan', '666884444', '1962-09-15', '975 Fire Oak, Humble, TX',
'M', '38000', '333445555', 5),
('James', 'E', 'Borg', '888665555', '1937-11-10', '450 Stone, Houston, TX', 'M',
'55000', NULL, 1),

```



```

('Jennifer', 'S', 'Wallace', '987654321', '1941-06-20', '291 Berry, Bellaire,
TX', 'F', '43000', '888665555', 4),
('Ahmad', 'V', 'Jabbar', '987987987', '1969-03-29', '980 Dallas, Houston, TX',
'M', '25000', '987654321', 4),
('Alicia', 'J', 'Zelaya', '999887777', '1968-07-19', '3321 Castle, Spring, TX',
'F', '25000', '987654321', 4);

```

```

-- -----

```

```

--

```

```

-- Struttura della tabella `LAVORA_SU`

```

```

--

```

```

CREATE TABLE `LAVORA_SU` (
  `SSN_I` char(9) NOT NULL,
  `N_P` int(11) NOT NULL,
  `ORE` decimal(3,1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--

```

```

-- Dump dei dati per la tabella `LAVORA_SU`

```

```

--

```

```

INSERT INTO `LAVORA_SU` (`SSN_I`, `N_P`, `ORE`) VALUES
('123456789', 1, '32.5'),
('123456789', 2, '7.5'),
('333445555', 2, '10.0'),
('333445555', 3, '10.0'),
('333445555', 10, '10.0'),
('333445555', 20, '10.0'),
('453453453', 1, '20.0'),
('453453453', 2, '20.0'),
('666884444', 3, '40.0'),
('888665555', 20, '0.0'),
('987654321', 20, '15.0'),
('987654321', 30, '20.0'),
('987987987', 10, '35.0'),
('987987987', 30, '5.0'),
('999887777', 10, '10.0'),
('999887777', 30, '30.0');

```

```

-- -----

```

```

--

```

```

-- Struttura della tabella `PERSONA_A_CARICO`
--

CREATE TABLE `PERSONA_A_CARICO` (
  `SSN_I` char(9) NOT NULL,
  `NOME_PERSONA_A_CARICO` varchar(15) NOT NULL,
  `SESSO` char(1) DEFAULT NULL,
  `DATA_N` date DEFAULT NULL,
  `PARENTELA` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dump dei dati per la tabella `PERSONA_A_CARICO`
--

INSERT INTO `PERSONA_A_CARICO` (`SSN_I`, `NOME_PERSONA_A_CARICO`, `SESSO`,
`DATA_N`, `PARENTELA`) VALUES
('123456789', 'Alice', 'F', '1988-12-30', 'Figlia'),
('123456789', 'Elizabeth', 'F', '1967-05-05', 'Coniuge'),
('123456789', 'Michael', 'M', '1988-01-04', 'Figlio'),
('333445555', 'Alice', 'F', '1986-04-05', 'Figlia'),
('333445555', 'Joy', 'F', '1958-05-03', 'Coniuge'),
('333445555', 'Theodore', 'M', '1983-10-25', 'Figlio'),
('987654321', 'Abner', 'M', '1942-02-28', 'Coniuge');

-----

--

-- Struttura della tabella `PROGETTO`
--

CREATE TABLE `PROGETTO` (
  `NOME_P` varchar(20) NOT NULL,
  `NUMERO_P` int(11) NOT NULL,
  `SEDE_P` varchar(15) DEFAULT NULL,
  `NUM_D` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dump dei dati per la tabella `PROGETTO`
--

INSERT INTO `PROGETTO` (`NOME_P`, `NUMERO_P`, `SEDE_P`, `NUM_D`) VALUES
('ProdottoX', 1, 'Bellaire', 5),

```

```

('ProdottoY', 2, 'Sugarland', 5),
('ProdottoZ', 3, 'Houston', 5),
('Informatizzazione', 10, 'Stafford', 4),
('Riorganizzazione', 20, 'Houston', 1),
('Nuove opportunità', 30, 'Stafford', 4);

-- -----

--
-- Struttura della tabella `SEDI_DIP`
--

CREATE TABLE `SEDI_DIP` (
  `NUMERO_D` int(11) NOT NULL,
  `SEDE_D` varchar(15) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dump dei dati per la tabella `SEDI_DIP`
--

INSERT INTO `SEDI_DIP` (`NUMERO_D`, `SEDE_D`) VALUES
(1, 'Houston'),
(4, 'Stafford'),
(5, 'Bellaire'),
(5, 'Houston'),
(5, 'Sugarland');

--
-- Indici per le tabelle scaricate
--

--
-- Indici per le tabelle `DIPARTIMENTO`
--

ALTER TABLE `DIPARTIMENTO`
  ADD PRIMARY KEY (`NUMERO_D`),
  ADD UNIQUE KEY `NOME_D` (`NOME_D`),
  ADD KEY `SSN_DIR` (`SSN_DIR`);

--
-- Indici per le tabelle `IMPIEGATO`
--

ALTER TABLE `IMPIEGATO`

```

```

ADD PRIMARY KEY (`SSN`),
ADD KEY `N_D` (`N_D`),
ADD KEY `SUPER_SSN` (`SUPER_SSN`);

--
-- Indici per le tabelle `LAVORA_SU`
--
ALTER TABLE `LAVORA_SU`
  ADD PRIMARY KEY (`SSN_I`,`N_P`),
  ADD KEY `N_P` (`N_P`);

--
-- Indici per le tabelle `PERSONA_A_CARICO`
--
ALTER TABLE `PERSONA_A_CARICO`
  ADD PRIMARY KEY (`SSN_I`,`NOME_PERSONA_A_CARICO`);

--
-- Indici per le tabelle `PROGETTO`
--
ALTER TABLE `PROGETTO`
  ADD PRIMARY KEY (`NUMERO_P`),
  ADD UNIQUE KEY `NOME_P` (`NOME_P`),
  ADD KEY `NUM_D` (`NUM_D`);

--
-- Indici per le tabelle `SEDI_DIP`
--
ALTER TABLE `SEDI_DIP`
  ADD PRIMARY KEY (`NUMERO_D`,`SEDE_D`);

--
-- Limiti per le tabelle scaricate
--

--
-- Limiti per la tabella `DIPARTIMENTO`
--
ALTER TABLE `DIPARTIMENTO`
  ADD CONSTRAINT `DIPARTIMENTO_ibfk_1` FOREIGN KEY (`SSN_DIR`) REFERENCES
`IMPIEGATO` (`SSN`) ON UPDATE CASCADE;

--
-- Limiti per la tabella `IMPIEGATO`

```

```

--
ALTER TABLE `IMPIEGATO`
  ADD CONSTRAINT `IMPIEGATO_ibfk_1` FOREIGN KEY (`N_D`) REFERENCES `DIPARTIMENTO`
(`NUMERO_D`) ON UPDATE CASCADE,
  ADD CONSTRAINT `IMPIEGATO_ibfk_2` FOREIGN KEY (`SUPER_SSN`) REFERENCES
`IMPIEGATO` (`SSN`) ON DELETE SET NULL ON UPDATE CASCADE;

--
-- Limiti per la tabella `LAVORA_SU`
--
ALTER TABLE `LAVORA_SU`
  ADD CONSTRAINT `LAVORA_SU_ibfk_1` FOREIGN KEY (`SSN_I`) REFERENCES `IMPIEGATO`
(`SSN`) ON UPDATE CASCADE,
  ADD CONSTRAINT `LAVORA_SU_ibfk_2` FOREIGN KEY (`N_P`) REFERENCES `PROGETTO`
(`NUMERO_P`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Limiti per la tabella `PERSONA_A_CARICO`
--
ALTER TABLE `PERSONA_A_CARICO`
  ADD CONSTRAINT `PERSONA_A_CARICO_ibfk_1` FOREIGN KEY (`SSN_I`) REFERENCES
`IMPIEGATO` (`SSN`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Limiti per la tabella `PROGETTO`
--
ALTER TABLE `PROGETTO`
  ADD CONSTRAINT `PROGETTO_ibfk_1` FOREIGN KEY (`NUM_D`) REFERENCES
`DIPARTIMENTO` (`NUMERO_D`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Limiti per la tabella `SEDI_DIP`
--
ALTER TABLE `SEDI_DIP`
  ADD CONSTRAINT `SEDI_DIP_ibfk_1` FOREIGN KEY (`NUMERO_D`) REFERENCES
`DIPARTIMENTO` (`NUMERO_D`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

## 5) Interrogazioni SQL

- a. Si crei un elenco di tutti i numeri di progetto dei progetti che coinvolgono un dipendente il cui cognome è 'Wong', come partecipante oppure come dirigente del dipartimento che controlla il progetto.

```
SELECT DISTINCT NUMERO_P
FROM PROGETTO
WHERE NUMERO_P IN
    (SELECT NUMERO_P
     FROM PROGETTO, DIPARTIMENTO, IMPIEGATO
     WHERE NUM_D=NUMERO_D AND
          SSN_DIR=SSN AND COGNOME = 'Wong')
OR NUMERO_P IN
    (SELECT N_P
     FROM LAVORA_SU, IMPIEGATO
     WHERE SSN_I=SSN AND COGNOME = 'Wong');
```

- b. Si trovi il nome di ciascun dipendente che ha una persona a carico con lo stesso sesso del dipendente.

```
SELECT I.NOME_BATT, I.COGNOME
FROM IMPIEGATO AS I
WHERE I.SSN IN
    (SELECT SSN_I
     FROM PERSONA_A_CARICO AS C
     WHERE I.SESSO = C.SESSO);
```

- c. Seleziona gli impiegati che hanno più di 2 persone a carico.

```
SELECT COGNOME, NOME_BATT
FROM IMPIEGATO
WHERE (SELECT COUNT(*)
      FROM PERSONA_A_CARICO
      WHERE SSN=SSN_I) >= 2;
```

- d. Per ciascun progetto in cui lavorano più di due impiegati si trovino il numero del progetto, il nome del progetto e il numero di impiegati che vi lavorano.

```
SELECT NUMERO_P, NOME_P, COUNT(*)
FROM PROGETTO, LAVORA_SU
WHERE NUMERO_P = N_P
GROUP BY NUMERO_P, NOME_P
HAVING COUNT(*) > 2;
```