1. Selezionare tutti gli insegnanti

SELECT \*

FROM `teachers`;

1. Selezionare tutti i referenti per ogni dipartimento

SELECT `id`,`name`,`head\_of\_departments`

FROM `departments`;

1. Selezionare tutti gli studenti il cui nome inizia per "E" (373)

SELECT `id`,`name`,`surname`

FROM `students`

WHERE `name` LIKE “E%”;

1. Selezionare tutti gli studenti che si sono iscritti nel 2021 (734)

SELECT \*

FROM `students`

WHERE `enrolment\_date` BETWEEN “2021-01-01” AND “2021-12-31”;

1. Selezionare tutti i corsi che non hanno un sito web (676)

SELECT `id`,`name`

FROM `courses`

WHERE `website` IS NULL;

1. Selezionare tutti gli insegnanti che hanno un numero di telefono (50)

SELECT `id`,`name`

FROM `teachers`

WHERE `phone` IS NOT NULL;

1. Selezionare tutti gli appelli d'esame dei mesi di giugno e luglio 2020 (2634)

SELECT \*

FROM `exams`

WHERE `date` BETWEEN “2020-06-01” AND “2020-07-31”;

1. Qual è il numero totale degli studenti iscritti? (5000)

SELECT COUNT(`id`) AS `total\_students`

FROM `students`;

2.1. Contare i corsi raggruppati per cfu

SELECT COUNT(`id`) AS `num\_courses`,`cfu`

FROM `courses`

GROUP BY `cfu`

HAVING `numbers\_courses`> 95;

2.2. Contare gli studenti raggruppati per anno di nascita

SELECT COUNT(`id`) AS `tot\_students`, YEAR(`date\_of\_birth`) AS `birth\_year`

FROM `students`

GROUP BY `birth\_year`;

2.3. Selezionare il voto più basso dato ad ogni appello d'esame

SELECT `exam\_id`, MIN(`vote`) AS `vote\_min`

FROM `exam\_student`

GROUP BY `exam\_id`;

2.4. Contare gli appelli d'esame del mese di luglio raggruppati per giorno

SELECT COUNT(`id`) AS `num\_appelli`, DAY(`date`) AS `day`

FROM `exams`

WHERE MONTH(`date`) = 7

GROUP BY DAY(`date`);

3.1. Contare quanti iscritti ci sono stati ogni anno

SELECT

COUNT(`enrolment\_date`) AS `iscrizioni`,

YEAR(`enrolment\_date`) AS `anno`

FROM `students`

GROUP BY `anno`

ORDER BY(`anno`) ASC;

3.2. Contare gli insegnanti che hanno l'ufficio nello stesso edificio

SELECT

office\_address,

COUNT(`name`)

FROM teachers

GROUP BY `office\_address`;

3.3. Calcolare la media dei voti di ogni appello d'esame

SELECT

exams.location,

`exam\_id` as `esame`,

AVG(`vote`) AS `media\_voti`

FROM exam\_student

INNER JOIN exams ON exam\_student.exam\_id = exams.id

GROUP BY `esame`;

3.4. Contare quanti corsi di laurea ci sono per ogni dipartimento

SELECT

departments.name,

COUNT(courses.id)

FROM courses

INNER JOIN degrees ON courses.degree\_id = degrees.id

INNER JOIN departments ON degrees.id = departments.id

GROUP BY departments.name;

4.1. Selezionare tutti gli studenti nati nel 1990 (160)

SELECT \*

FROM students

WHERE YEAR(date\_of\_birth)= 1990;

4.2. Selezionare tutti i corsi che valgono più di 10 crediti (479)

SELECT \*

FROM courses

WHERE cfu > 10;

4.3. Selezionare tutti gli studenti che hanno più di 30 anni

SELECT \*

FROM `students`

WHERE YEAR(current\_date()) - YEAR(date\_of\_birth) > 30

ORDER BY `date\_of\_birth` DESC;

SELECT \*

FROM `students`

WHERE TIMESTAMPDIFF(YEAR, date\_of\_birth, CURDATE()) > 30

ORDER BY `date\_of\_birth` DESC;

4.4. Selezionare tutti i corsi del primo semestre del primo anno di un qualsiasi corso di  
laurea (286)

SELECT \*

FROM courses

WHERE `year` = 1

AND `period` = "I semestre";

4.5. Selezionare tutti gli appelli d'esame che avvengono nel pomeriggio (dopo le 14) del  
20/06/2020 (21)

SELECT \*

FROM `exams`

WHERE TIME(hour) > '14:00:00'

AND DATE = '2020-06-20'

ORDER BY `hour` ASC;

4.6. Selezionare tutti i corsi di laurea magistrale (38)

SELECT \*

FROM `degrees`

WHERE `level` = "magistrale";

4.7. Da quanti dipartimenti è composta l'università? (12)

SELECT

COUNT(`id`) AS `total\_departments`

FROM `departments`;

4.8. Quanti sono gli insegnanti che non hanno un numero di telefono? (50)

SELECT

COUNT(\*)

FROM `teachers`

WHERE phone IS NOT NULL;

4.9. Inserire nella tabella degli studenti un nuovo record con i propri dati (per il campo  
degree\_id, inserire un valore casuale)

INSERT INTO `students` (degree\_id,name,surname,date\_of\_birth,fiscal\_code,enrolment\_date,registration\_number,email)

VALUES (1,"Gigi","Gigel","1997-08-03","giggg98m03z132f","2021-03-12","97316","ciaogigi@sonoio.com");

SELECT \*

FROM `students`

WHERE name = 'Gigi';

4.10. Cambiare il numero dell’ufficio del professor Pietro Rizzo in 126

UPDATE `teachers`

SET `office\_number` = 126

WHERE `id` = 125;

4.11. Eliminare dalla tabella studenti il record creato precedentemente al punto 9

DELETE FROM `students`

WHERE `degree\_id` = 1 AND `name` = 'Gigi';

5.2. Selezionare le informazioni sul corso con id = 144, con tutti i relativi appelli d’esame

SELECT `courses`.\*,`exams`.\*

FROM `courses`

INNER JOIN `exams` ON `courses`.id = `exams`.`course\_id`

WHERE `course\_id` = 144;

5.3. Selezionare a quale dipartimento appartiene il Corso di Laurea in Diritto  
dell'Economia (Dipartimento di Scienze politiche, giuridiche e studi internazionali)

SELECT `departments`.\*, `degrees`.`name`

FROM `departments`

INNER JOIN `degrees`

ON `degrees`.`department\_id` = `departments`.`id`

WHERE `departments`.`name` = "Dipartimento di Scienze politiche, giuridiche e studi internazionali";

5.4. Selezionare tutti gli appelli d'esame del Corso di Laurea Magistrale in Fisica del  
primo anno

5.5. Selezionare tutti i docenti che insegnano nel Corso di Laurea in Lettere (21)

SELECT DISTINCT `teachers`.\*

FROM `degrees`

JOIN `courses` ON `degrees`.`id` = `courses`.`degree\_id`

JOIN `course\_teacher` ON `course\_teacher`.`course\_id` = `courses`.`id`

JOIN `teachers` ON `course\_teacher`.`teacher\_id` = `teachers`.`id`

WHERE `degrees`.`name` = "Corso di Laurea in Lettere";

5.6. Selezionare il libretto universitario di Mirco Messina (matricola n. 620320)

SELECT `students`.`id`, `students`.`name`, `students`.`surname`,`courses`.`name`,`exam\_student`.`vote`

FROM students

INNER JOIN `exam\_student` ON `exam\_student`.`student\_id` = `students`.`id`

INNER JOIN `exams` ON `exams`.`id` = `exam\_student`.`exam\_id`

INNER JOIN `courses` ON `courses`.`id` = `exams`.`course\_id`

WHERE `students`.`registration\_number` = 620320

AND `students`.`name` = "Mirco"

AND `students`.`surname` = "Messina"

AND `exam\_student`.`vote` >= 18;

5.7. Selezionare il voto medio di superamento d'esame per ogni corso, con anche i dati  
del corso di laurea associato, ordinati per media voto decrescente

SELECT `courses`.`name`, AVG(`exam\_student`.`vote`) AS `media\_voti`,`degrees`.`name`

FROM `exam\_student`

INNER JOIN `exams` ON `exam\_student`.`exam\_id` = `exams`.`id`

INNER JOIN `courses` ON `exams`.`course\_id` = `courses`.`id`

INNER JOIN `degrees` ON `courses`.`degree\_id` = `degrees`.`id`

WHERE `exam\_student`.`vote` >= 18

GROUP BY `courses`.`id`,`degrees`.`id`

ORDER BY `media\_voti` DESC;