# BUSINESS INTELLIGENCE A.A. 2023-2024



Gianni Eduard Balbin Canchanya - 901609

Alberto Invernizzi - 901611

Distribuzione del numero degli studenti iscritti nei vari appelli, suddivisa per anni e per corso di laurea

```
SELECT appelli.dtappello,
   appelli.adcod,
                                                                       SELECT DtAppello,
   ad.ad,
                                                                          AdCod,
   appelli.annoAcc,
                                                                          AD,
   count(studente) AS totstudenti,
                                                                          annoAcc,
   appelli.appcod
                                                                          count(studente) AS totstudenti
FROM appelli
                                                                        FROM bos_denormalizzato
   JOIN iscrizioni ON appelli.appcod = iscrizioni.appcod
                                                                       GROUP BY DtAppello,
   JOIN ad ON appelli.adcod = ad.adcod
                                                                            AdCod
WHERE dtappello IS NOT NULL
                                                                       ORDER BY annoAcc ASC;
GROUP BY appelli.appcod
ORDER BY appelli.dtappello;
```



	dtappello	adcod	ad	annoAcc	totstudenti	appcod
1	2015-09-17	E2401P013	[E2401P013] PSICOLOGIA DINAMICA	2014/2015	1	3459
2	2016-01-20	F5102P015	[F5102P015] PSICOLOGIA DELLE INFLUENZE SOCIALI	2015/2016	1	6860
3	2016-01-25	E2401P103	[E2401P103] PSICOPATOLOGIA GENERALE	2015/2016	1	3605
4	2016-01-25	F5102P016	[F5102P016] ATTEGGIAMENTI E OPINIONI	2015/2016	1	6865
5	2016-01-25	F5102P019	[F5102P019] PSICOLOGIA DELLE CONDOTTE FINANZIARIE	2015/2016	1	6877
6	2016-02-24	F5102P012	[F5102P012] COMUNICAZIONE NELLE ORGANIZZAZIONE E COMUNICAZIONE SOCIALE	2015/2016	1	6847
7	2016-02-26	F5101P139	[F5101P139] PSICOLOGIA DELLA SALUTE E INTERVENTI CLINICI IN AMBITO SANITARIO	2015/2016	1	6774
8	2016-04-26	E2702Q099	[E2702Q099] CHIMICA FISICA APPLICATA	2015/2016	1	3917
9	2016-05-31	E2702Q097	[E2702Q097] CHIMICA FISICA III E LABORATORIO	2015/2016	1	3909
10	2016-06-10	F5102P004	[F5102P004] COGNIZIONE SOCIALE	2015/2016	2	6828

	DtAppello	AdCod	AD	annoAcc	totstudenti
1	2015-09-17	E2401P013	[E2401P013] PSICOLOGIA DINAMICA	2014/2015	1
2	2016-01-08	F5102P019	[F5102P019] PSICOLOGIA DELLE CONDOTTE FINANZIARIE	2015/2016	1
3	2016-01-20	F5102P015	[F5102P015] PSICOLOGIA DELLE INFLUENZE SOCIALI	2015/2016	1
4	2016-01-25	E2401P103	[E2401P103] PSICOPATOLOGIA GENERALE	2015/2016	1
5	2016-01-25	F5102P016	[F5102P016] ATTEGGIAMENTI E OPINIONI	2015/2016	1
6	2016-01-25	F5102P019	[F5102P019] PSICOLOGIA DELLE CONDOTTE FINANZIARIE	2015/2016	1
7	2016-02-05	F5103P002	[F5103P002] METODI DI RICERCA E VALUTAZIONE IN PSICOLOGIA DELLO SVILUPPO	2015/2016	1
8	2016-02-24	F5102P012	[F5102P012] COMUNICAZIONE NELLE ORGANIZZAZIONE E COMUNICAZIONE SOCIALE	2015/2016	1
9	2016-02-26	F5101P139	[F5101P139] PSICOLOGIA DELLA SALUTE E INTERVENTI CLINICI IN AMBITO SANITARIO	2015/2016	1
10	2016-04-26	E2702Q099	[E2702Q099] CHIMICA FISICA APPLICATA	2015/2016	1

For this query, we have decided to add an attribute in both *appelli* and *bos\_denormalizzato*. Specifically, we created the academic year column (annoAcc), and based on a condition, a certain value was assigned. For instance, if 'dtappello' is between '2014-09-25' and '2015-09-24,' the value of 'annoAcc' will be '2014/2015,' and so on.

No significant differences were observed, except that in *appelli*, there are some NULL values in the 'dtappello' attribute, which is not found in *bos\_denormalizzato*.

Therefore, we are not surprised if bos\_denormalizzato returns more observations (293 observations).

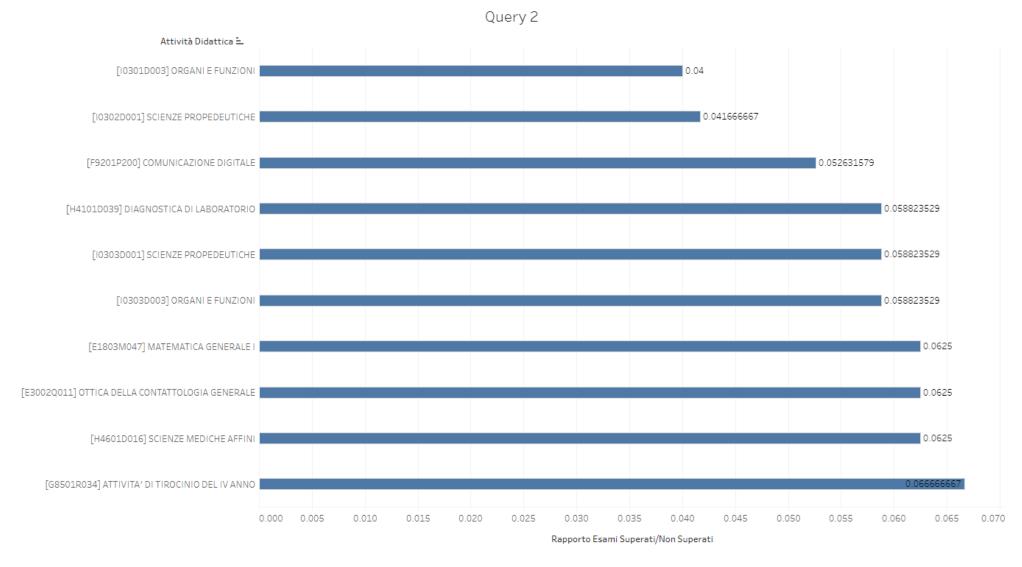


Individuazione della Top-10 degli esami più difficili suddivisi per corso di studi.

```
SELECT tl.appcod,
   tl.ad,
   tl.totSup.
   t2.totNoSup,
   (t1.totSup * 1.0 / (t1.totSup + t2.totNoSup) ) AS ratio
FROM (
     SELECT appelli.appcod,
         ad.ad.
         count(appelli.appcod) AS totSup
      FROM iscrizioni
         JOIN appelli ON iscrizioni.appcod = appelli.appcod
         [OIN ad ON appelli.adcod = ad.adcod
      WHERE assenza = 0 AND Superamento = 1
      GROUP BY appelli.appcod
   ) AS tl
   JOIN
     SELECT appelli.appcod,
         ad.ad.
         count(appelli.appcod) AS totNoSup
      FROM iscrizioni
         JOIN appelli ON iscrizioni.appcod = appelli.appcod
         IOIN ad ON appelli.adcod = ad.adcod
      WHERE assenza = 0 AND Superamento = 0
      GROUP BY appelli.appcod
   ) AS t2 ON t1.appcod = t2.appcod
ORDER BY ratio ASC
LIMIT 10:
```

```
SELECT tl.DtAppello,
   tl.AD.
   tl.totSup,
   t2.totNoSup,
   (t1.totSup * 1.0 / (t1.totSup + t2.totNoSup) ) AS ratio
FROM (
     SELECT AD,
        DtAppello,
        Count(DtAppello) AS totSup
      FROM bos denormalizzato
     WHERE Assenza = 0 AND Superamento = 1
     GROUP BY DtAppello, AD
     ORDER BY AD ASC
   ) AS tl
   IOIN
     SELECT AD.
        DtAppello.
        Count(DtAppello) AS totNoSup
      FROM bos denormalizzato
     WHERE Assenza = 0 AND Superamento = 0
     GROUP BY DtAppello, AD
     ORDER BY AD ASC
   ) AS t2 ON t1.DtAppello = t2.DtAppello AND t1.AD = t2.AD
ORDER BY ratio ASC
LIMIT 10;
```





There are no relevant differences for both queries.



Individuazione dei corsi di laurea ad elevato tasso di commitment, ovvero appelli di esami diversi ma del medesimo corso di laurea che si sono svolti nello stesso giorno

```
SELECT cds.cds,
appelli.dtappello,
count(cds.cds) AS esamiSvolti
FROM appelli
JOIN
cds ON appelli.cdscod = cds.cdscod
WHERE appelli.dtappello IS NOT NULL
GROUP BY appelli.dtappello,
appelli.cdscod
ORDER BY esamiSvolti DESC;
```

SELECT Cds,
DtAppello,
count(DISTINCT AD) AS
EsamiSvolti
FROM bos\_denormalizzato
GROUP BY DtAppello,
Cds
ORDER BY EsamiSvolti DESC;



	cds	dtappello	esamiSvolti
1	[H4101D] MEDICINA E CHIRURGIA	2017-03-31	42
2	[581] GIURISPRUDENZA	2017-01-25	24
3	[581] GIURISPRUDENZA	2016-11-30	23
4	[581] GIURISPRUDENZA	2017-09-27	23
5	[581] GIURISPRUDENZA	2016-11-28	19
6	[581] GIURISPRUDENZA	2017-07-19	19
7	[581] GIURISPRUDENZA	2017-09-11	19
8	[581] GIURISPRUDENZA	2017-11-27	19
9	[581] GIURISPRUDENZA	2017-11-29	18
10	[I0101D] INFERMIERISTICA (ABILITANTE ALLA PROFESSIONE SANITARIA DI INFERMIERE)	2017-09-22	17

	Cds	DtAppello	EsamiSvolti
1	[H4101D] MEDICINA E CHIRURGIA	2017-03-31	42
2	[581] GIURISPRUDENZA	2017-05-03	30
3	[581] GIURISPRUDENZA	2017-01-25	24
4	[581] GIURISPRUDENZA	2016-11-30	22
5	[581] GIURISPRUDENZA	2017-05-04	21
6	[581] GIURISPRUDENZA	2017-09-27	21
7	[581] GIURISPRUDENZA	2016-11-28	19
8	[581] GIURISPRUDENZA	2017-02-08	19
9	[581] GIURISPRUDENZA	2017-09-11	18
10	[581] GIURISPRUDENZA	2017-11-27	18

As with query 1, for this query, we also observed a slight difference in 'esamiSvolti', probably due to the fact that in the first query, we excluded NULL values from 'dtappello'.



ORDER BY t4.cds, t4.media;

Individuazione della Top-3 degli esami con media voti maggiore e minore rispettivamente, calcolati per ogni singolo corso di studi

```
SELECT DISTINCT t4.ad, t4.cds, t4.media
FROM (
     SELECT t3.ad, t3.cds, t3.media
      FROM (
                                                                                           SELECT DISTINCT t4.AD, t4.CdS, t4.media
          SELECT *, row_number() OVER (PARTITION BY t2.cds ORDER BY t2.cds) AS row_num
                                                                                            FROM (
           FROM (
                                                                                                 SELECT t3.AD, t3.CdS, t3.media
                SELECT *
                                                                                                 FROM (
                 FROM (SELECT ad.ad, cds.cds, avg(iscrizioni.Voto) AS media
                                                                                                      SELECT *, row number() OVER (PARTITION BY t2.CdS ORDER BY t2.CdS) AS row num
                      FROM iscrizioni
                                                                                                       FROM (
                         JOIN appelli ON iscrizioni.appcod = appelli.appcod
                                                                                                           SELECT *
                         JOIN ad ON appelli.adcod = ad.adcod
                                                                                                            FROM (SELECT AD, CdS, avg(Voto) AS media
                         JOIN cds ON appelli.cdscod = cds.cdscod
                                                                                                                  FROM bos denormalizzato
                      WHERE iscrizioni. Voto IS NOT NULL
                                                                                                                 WHERE Voto IS NOT NULL
                      GROUP BY ad.ad, cds.cds
                                                                                                                  GROUP BY AD, Cds
                      ORDER BY media ASC
                                                                                                                 ORDER BY media ASC
                   ) AS t1 ORDER BY t1.cds
                                                                                                               ) AS t1 ORDER BY t1.CdS
              ) AS t2 ) AS t3
                                                                                                         ) AS t2 ) AS t3
     WHERE t3.row num <= 3
                                                                                                 WHERE t3.row num <= 3
     UNION ALL
                                                                                                 UNION ALL
     SELECT t3.ad, t3.cds, t3.media
                                                                                                 SELECT t3.AD, t3.CdS, t3.media
      FROM (
                                                                                                 FROM (
          SELECT *, row number() OVER (PARTITION BY t2.cds ORDER BY t2.cds DESC) AS row num
                                                                                                      SELECT *, row number() OVER (PARTITION BY t2.CdS ORDER BY t2.CdS DESC) AS row num
           FROM (
                                                                                                       FROM (
                SELECT *
                                                                                                           SELECT *
                 FROM (SELECT ad.ad, cds.cds, avg(iscrizioni.Voto) AS media
                                                                                                            FROM (SELECT AD, CdS, avg(Voto) AS media
                      FROM iscrizioni
                                                                                                                  FROM bos denormalizzato
                         JOIN appelli ON iscrizioni.appcod = appelli.appcod
                                                                                                                 WHERE Voto IS NOT NULL
                         JOIN ad ON appelli.adcod = ad.adcod
                                                                                                                 GROUP BY AD, Cds
                         JOIN cds ON appelli.cdscod = cds.cdscod
                                                                                                                  ORDER BY media DESC
                      WHERE iscrizioni. Voto IS NOT NULL
                                                                                                               ) AS t1 ORDER BY t1.CdS
                      GROUP BY ad.ad, cds.cds
                                                                                                         ) AS t2 ) AS t3
                      ORDER BY media DESC
                                                                                                 WHERE t3.row num <= 3
                   ) AS t1 ORDER BY t1.cds
                                                                                                 ORDER BY t3.CdS
              ) AS t2 ) AS t3
                                                                                              ) AS t4
     WHERE t3.row num <= 3
                                                                                            ORDER BY t4.CdS, t4.media;
      ORDER BY t3.cds
   ) AS t4
```

#### Query 4

CorsoDiStudi	Attività Didattica	≐
[524] SCIENZE DELL'EDUCAZIONE	[E1901R086] METODI E TECNICHE DELLA FORMAZIONE MEDIALE	25,00
[541] SCIENZE DEL TURISMO E COMUNITA' LOCALE(C.SO A DISTANZA)	[E1501N009] SOCIOLOGIA DELLA CULTURA E DEL TEMPO LIBERO	25,00
	[A5810081] ECONOMIA AZIENDALE	21,00
	[A5810001] DIRITTO DEI CONSUMATORI	23,67
[581] GIURISPRUDENZA	[A5810130] DIRITTO DELL'UNIONE EUROPEA	25,10
[201] GIOKISPKODENZA	[A5810139] DIRITTO AMMINISTRATIVO -PARTE SPECIALE	30,00
	[A5810169] EUROPEAN UNION COMMON AND SECURITY DEFENSE POLICY	30,00
	[A5810170] TRANSITIONAL JUSTICE AND STATE BUILDING	30,00
	[E0201Q050] METODOLOGIE BIOCHIMICHE E TECNOLOGIE BIOMOLECOLARI	22,63
	[E1301Q059] ECOLOGIA APPLICATA	23,00
	[E0201Q059] BIOCHIMICA PER LE BIOTECNOLOGIE	23,41
CorsoDiStudi	Query 4 Denormalizzato  Attività Didattica	<u>.</u>
[524] SCIENZE DELL'EDUCAZIONE	[E1901R086] METODI E TECNICHE DELLA FORMAZIONE MEDIALE	25,00
[541] SCIENZE DEL TURISMO E COMUNITA' LOCALE(C.SO A DISTANZA)	[E1501N009] SOCIOLOGIA DELLA CULTURA E DEL TEMPO LIBERO	25,00
	[A5810081] ECONOMIA AZIENDALE	21,00
	[A5810001] DIRITTO DEI CONSUMATORI	23,67
[Foot] Guyanapayaraya	[A5810130] DIRITTO DELL'UNIONE EUROPEA	25,10
[581] GIURISPRUDENZA	[A5810139] DIRITTO AMMINISTRATIVO -PARTE SPECIALE	30,00
	[A5810169] EUROPEAN UNION COMMON AND SECURITY DEFENSE POLICY	30,00
	[A5810170] TRANSITIONAL JUSTICE AND STATE BUILDING	30,00
	[E0201Q050] METODOLOGIE BIOCHIMICHE E TECNOLOGIE BIOMOLECOLARI	22,63
	[E1301Q059] ECOLOGIA APPLICATA	23,00
	[E0201Q059] BIOCHIMICA PER LE BIOTECNOLOGIE	23,41

There are no relevant differences for both queries.



GROUP BY t3.cds;

Calcolare la distribuzione degli studenti "fast&furious" per corso di studi, ovvero studenti con il rapporto "votazione media riportata negli esami superati" su "periodo di attività" maggiore.

```
SELECT t3.cds, count(t3.cds) AS numStud
 FROM (
     SELECT t1.*, t2.media,
                                                                                              SELECT t3.cds, count(t3.cds) AS numStud
       (t2.media / CASE WHEN tl.diffGiorni = 0 THEN 1 ELSE tl.diffGiorni END) AS ratio
                                                                                               FROM (
      FROM (
                                                                                                    SELECT t1.*, t2.media,
           SELECT iscrizioni.studente, cds.cds,
                                                                                                        (t2.media / CASE WHEN t1.diffGiorni = 0 THEN 1 ELSE t1.diffGiorni END) AS ratio
              julianday(max(dtappello)) - julianday(min(dtappello)) AS diffGiorni
                                                                                                     FROM (
            FROM iscrizioni
                                                                                                          SELECT Studente, CdS,
              JOIN appelli ON iscrizioni.appcod = appelli.appcod
                                                                                                             julianday(max(DtAppello)) - julianday(min(DtAppello)) AS diffGiorni
              [OIN cds ON appelli.cdscod = cds.cdscod
                                                                                                          FROM bos denormalizzato
           WHERE appelli.dtappello IS NOT NULL
                                                                                                          GROUP BY Studente
            GROUP BY iscrizioni.studente
                                                                                                          ORDER BY CdS, Studente, DtAppello
            ORDER BY cds.cdscod, iscrizioni.studente, appelli.dtappello
                                                                                                       ) AS t1
         )AS t1
                                                                                                       IOIN
         IOIN
                                                                                                        (SELECT Studente, CdS, avg(Voto) AS media
         (SELECT iscrizioni.studente, cds.cds, avg(iscrizioni.Voto) AS media
                                                                                                          FROM bos denormalizzato
            FROM iscrizioni
                                                                                                          WHERE Voto IS NOT NULL
              [OIN appelli ON iscrizioni.appcod = appelli.appcod
                                                                                                          GROUP BY Studente
              [OIN cds ON appelli.cdscod = cds.cdscod
                                                                                                          ORDER BY CdS, Studente, DtAppello
           WHERE appelli.dtappello IS NOT NULL AND iscrizioni.Voto IS NOT NULL
                                                                                                       ) AS t2 ON t1.studente = t2.studente
            GROUP BY iscrizioni.studente
                                                                                                     WHERE ratio >= 0.73341637214748
            ORDER BY cds.cdscod, iscrizioni.studente, appelli.dtappello
                                                                                                     ORDER BY ratio DESC
         ) AS t2 ON t1.studente = t2.studente
                                                                                                  ) AS t3
      WHERE ratio >= 0.94350159714953
                                                                                               GROUP BY t3.cds;
      ORDER BY ratio DESC
   ) AS t3
```



#### Query 5

[581] GIURISPRUDENZA	[E1802M] ECONOMIA E AMMINISTRAZIONE	[E3501Q] MATEMATICA	[E3101Q] INFORMATICA	[E3001Q] FISICA	[E2401P] SCIENZE E	[F8701N]		
NUMERO STUDENTI: 104	DELLE IMPRESE	NUMERO STUDENTI: 32	IN ORMATICA	NUMERO	TECNICHE			
	NUMERO STUDENTI: 44	[F1801Q] INFORMATICA	[F0101R] SCIENZE ANTROPOLOGICHE ED	[E1301Q] SCIENZE BIOLOGICHE	[F1701Q] FISICA	[E0201Q]		
	[E1801M] MARKETING, COMUNICAZIONE	NUMERO STUDENTI: 31	ETNOLOGICHE	BIOLOGICILE	NUMERO			
	AZIENDALE E MERCATI GLOBALI	[E2001R] COMUNICAZIONE	[G8501R] SCIENZE DELLA FORMAZIONE	[F5701R] FORMAZIONE E	[E1401A] SCIENZE DEI	[E1501N] SCIENZE DEL		
[E1901R] SCIENZE DELL'EDUCAZIONE	[E4001N] SOCIOLOGIA	INTERCULTURALE	PRIMARIA	SVILUPPO DELLE	SERVIZI	[E2004P] SCIENZE		
NUMERO STUDENTI: 67	NUMERO STUDENTI: 36	[E1803M] ECONOMIA	[E1601N] SCIENZE	[F8501R] SCIENZE PEDAGOGICHE	[E3301M] ECONOMIA E	[F7701M]		
	TOMESO STORENTI. 30	DELLE BANCHE, DELLE ASSICURAZIONI E DEGLI			COMMERCIO	SCIENZE		

#### Query 5 Denormalizzato

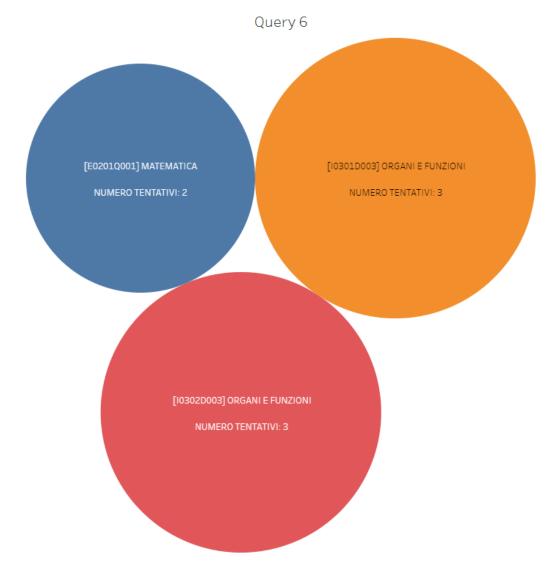
[581] GIURISPRUDENZA NUMERO STUDENTI: 58	[E1801M] MARKETING, COMUNICAZIONE AZIENDALE E MERCATI GLOBALI	[E3501Q] MATEMATICA NUMERO STUDENTI:	[E2001R] COMUNICAZIONE INTERCULTURALE	[F5701R] FORMAZIONE E SVILUPPO DELLE RISORSE UMANE	[G8501R] SCIENZE DELLA FORMAZIONE PRIMARIA	[E1401A] SCIENZE DEI SERVIZI	[E0201Q]		
			NUMERO STUDENTI:	[E1601N] SCIENZE	[E1803M]	[E3301M]	[E2004P] SCIENZE		
	[E1802M] ECONOMIA E AMMINISTRAZIONE DELLE IMPRESE	[E1301Q] SCIENZE BIOLOGICHE	[F8501R] SCIENZE PEDAGOGICHE	DELL'ORGANIZZAZIONE	ECONOMIA DELLE BANCHE, DELLE	ECONOMIA E COMMERCIO	30.21122		
[E1901R] SCIENZE DELL'EDUCAZIONE	NUMERO STUDENTI: 37	NUMERO STUDENTI:	NUMERO STUDENTI: 29	[E3001Q] FISICA	[E2401P] SCIENZE E TECNICHE	[F7701M] SCIENZE			
NUMERO STUDENTI: 51	[E4001N] SOCIOLOGIA	[E3101Q]	[F0101R] SCIENZE	NUMERO STUDENTI: 24	PSICOLOGICHE				
	NUMERO STUDENTI: 36	NUMERO STUDENTI:	ANTROPOLOGICHE ED ETNOLOGICHE	[F1701Q] FISICA  NUMERO STUDENTI: 24	[F1801Q] INFORMATICA	[E1501N] SCIENZE DEL TURISMO E			

In this query, we observed differences in results because bos\_denormalizzato has more observations, and for this reason, we adjusted the mean of the ratio between the two queries.



Individuazione della Top-3 degli esami "trial&error", ovvero esami che richiedono il maggior numero di tentativi prima del superamento.

```
SELECT t3.adcod, ad.ad, round(avg(t3.tot)) AS TrialAndError
FROM (SELECT *
                                                                                         SELECT t3.AdCod, AD, round(avg(t3.tot)) AS TrialAndError
      FROM (SELECT*, count(t1.studente) AS tot
                                                                                          FROM (SELECT *
            FROM (
                                                                                               FROM (SELECT*, count(tl.Studente) AS tot
                 SELECT iscrizioni.studente,
                                                                                                     FROM (
                    appelli.appcod,
                                                                                                         SELECT Studente.
                    appelli.adcod,
                                                                                                             DtAppello,
                    iscrizioni.Superamento,
                                                                                                             AdCod,
                    iscrizioni.Insufficienza,
                                                                                                             AD,
                    iscrizioni.Ritiro,
                                                                                                             Superamento.
                     iscrizioni.Voto
                                                                                                             Insufficienza,
                  FROM iscrizioni
                                                                                                             Ritiro,
                    JOIN appelli ON iscrizioni.appcod = appelli.appcod
                                                                                                             Voto
                 WHERE (iscrizioni.Superamento != 0 OR
                                                                                                          FROM bos_denormalizzato
                     iscrizioni.Insufficienza != 0 OR
                                                                                                          WHERE (Superamento != 0 OR
                     iscrizioni.Ritiro!= 0 OR
                                                                                                             Insufficienza!= 0 OR
                     iscrizioni. Voto IS NOT NULL)
                                                                                                             Ritiro != 0 OR
                                                                                                             Voto IS NOT NULL)
                 ORDER BY appelli.adcod,
                                                                                                          ORDER BY AdCod.
                      iscrizioni.studente.
                                                                                                              Studente.
                      appelli.appcod DESC
                                                                                                              DtAppello DESC
               ) AS tl
                                                                                                       ) AS tl
            GROUP BY tl.studente, tl.adcod
                                                                                                    GROUP BY t1.Studente, t1.AdCod
         ) AS t2
                                                                                                  ) AS t2
      ORDER BY t2.adcod
                                                                                               ORDER BY t2.AdCod
   ) AS t3
                                                                                            ) as t3
   [OIN ad ON t3.adcod = ad.adcod
                                                                                         GROUP BY t3.AdCod
GROUP BY t3.adcod
                                                                                         ORDER BY TrialAndError DESC, AD
ORDER BY TrialAndError DESC, ad.ad
                                                                                         LIMIT 3;
LIMIT 3;
```



There are no relevant differences for both queries.



ORDER BY tl.ad, Rapporto DESC;

Visualizza la media dei voti ottenuti dagli studenti iscritti agli esami per ciascuna attività didattica, considerando solo gli appelli con un tasso di superamento inferiore al 50%. Ordina i risultati in modo decrescente in base alla media dei voti.

```
SELECT tl.ad, tl.dtappello, tl.appcod, tl.Media_Voti,
   round(CAST (tl.Pass AS REAL) / CAST (tl.Tot AS REAL), 2) AS Rapporto
                                                                                    SELECT *
FROM (
     SELECT ad.ad.
                                                                                     FROM (
         appelli.dtappello,
                                                                                          SELECT AD,
         iscrizioni.appcod,
                                                                                             DtAppello,
        round(avg(iscrizioni.Voto), 2) AS Media_Voti,
                                                                                             round(avg(Voto), 2) AS Media_Voti,
         sum(iscrizioni.Superamento) AS Pass,
                                                                                             round(CAST (sum(Superamento) AS REAL) /
         sum(iscrizioni.Superamento + iscrizioni.Insufficienza + iscrizioni.Ritiro) AS Tot
                                                                                                  CAST (sum(Superamento + Insufficienza + Ritiro) AS REAL), 2) AS Rapporto
      FROM appelli
                                                                                           FROM bos denormalizzato
         IOIN ad ON appelli.adcod = ad.adcod
                                                                                          WHERE Assenza = 0
         JOIN iscrizioni ON appelli.appcod = iscrizioni.appcod
                                                                                          GROUP BY AD, DtAppello
     WHERE iscrizioni. Assenza = 0
                                                                                          ORDER BY AD
      GROUP BY iscrizioni.appcod
                                                                                        ) AS tl
     ORDER BY ad.ad
                                                                                     WHERE Rapporto < 0.5 AND Media Voti >= 18
   ) AS tl
                                                                                     ORDER BY AD, Rapporto DESC;
WHERE Rapporto < 0.5 AND Media_Voti >= 18
```



	ad	dtappello	appcod	Media_Voti	Rapporto
1	[A5810003] SCIENZA DELLE FINANZE	NULL	19	22.25	0.44
2	[A5810003] SCIENZA DELLE FINANZE	2017-07-21	21	23.43	0.44
3	[A5810010] DIRITTO AMMINISTRATIVO	NULL	51	27	0.33
4	[A5810010] DIRITTO AMMINISTRATIVO	NULL	53	25.5	0.25
5	[A5810015] DIRITTO PENALE	2017-07-11	100	25.25	0.42
6	[A5810017] DIRITTO PROCESSUALE CIVILE	NULL	119	25	0.33
7	[A5810019] ISTITUZIONI DI DIRITTO PRIVATO	2017-09-11	148	23.89	0.45
8	[A5810056] DIRITTO FALLIMENTARE	2017-07-19	325	28	0.33
9	[A5810057] DIRITTO INDUSTRIALE	NULL	337	23	0.33
10	[A5810062] DIRITTO PENALE MINORILE	2017-09-26	382	29	0.4

	AD	DtAppello	Media_Voti	Rapporto
1	[A5810003] SCIENZA DELLE FINANZE	2017-06-09	22.25	0.44
2	[A5810010] DIRITTO AMMINISTRATIVO	2017-02-08	27	0.33
3	[A5810010] DIRITTO AMMINISTRATIVO	2017-05-03	25.5	0.25
4	[A5810015] DIRITTO PENALE	2017-07-11	25.25	0.42
5	[A5810056] DIRITTO FALLIMENTARE	2017-07-19	28	0.33
6	[A5810057] DIRITTO INDUSTRIALE	2017-05-04	23	0.33
7	[A5810062] DIRITTO PENALE MINORILE	2017-09-26	29	0.4
8	[E0201Q001] MATEMATICA	2017-02-22	22.56	0.26
9	[E0201Q001] MATEMATICA	2017-09-13	26	0.11
10	[E0201Q005] CHIMICA ORGANICA	2017-10-16	24.2	0.42

In this query, bos\_denormalizzato returns fewer observations (401 observations) because in the first query, it excludes some observations that, if included, would give me a ratio greater than 0.5. In fact, in the first query with 452 observations, two instances of 'Scienza delle Finanze' are noticeable, one of which has the same 'Media\_Voti' and the same 'Rapporto'.

