```
{
  "Spider dataset": {
    "statistics": "For the Spider dataset, we corrected the gold SQL for 27 ques
tions. The original gold SQL is marked with the field 'ori_gold', and the revise
d gold SQL is marked with the field 'revised_gold'.",
    "details": [
      {
        "question_index": 11,
        "db_id": "pets_1",
        "question": "What is the first name of every student who has a dog but d
oes not have a cat?",
        "ori_gold": "SELECT T1.fname ,  T1.age FROM student AS T1 JOIN has_pet A
S T2 ON T1.stuid  =  T2.stuid JOIN pets AS T3 ON T3.petid  =  T2.petid WHERE T3.
pettype  =  'dog' AND T1.stuid NOT IN (SELECT T1.stuid FROM student AS T1 JOIN h
as_pet AS T2 ON T1.stuid  =  T2.stuid JOIN pets AS T3 ON T3.petid  =  T2.petid W
HERE T3.pettype  =  'cat')",
        "revised_gold": "SELECT T1.fname FROM student AS T1 JOIN has_pet AS T2 O
N T1.stuid  =  T2.stuid JOIN pets AS T3 ON T3.petid  =  T2.petid WHERE T3.pettyp
e  =  'dog' AND T1.stuid NOT IN (SELECT T1.stuid FROM student AS T1 JOIN has_pet
 AS T2 ON T1.stuid  =  T2.stuid JOIN pets AS T3 ON T3.petid  =  T2.petid WHERE T
3.pettype  =  'cat')"
      },
      {
        "question_index": 46,
        "db_id": "flight_2",
        "question": "What is the code of airport that has fewest number of fligh
ts?",
        "ori_gold": "SELECT T1.AirportCode FROM AIRPORTS AS T1 JOIN FLIGHTS AS T
2 ON T1.AirportCode  =  T2.DestAirport OR T1.AirportCode  =  T2.SourceAirport GR
OUP BY T1.AirportCode ORDER BY count(*) LIMIT 1",
        "revised_gold": "SELECT T1.AirportCode FROM AIRPORTS AS T1 LEFT JOIN FLI
GHTS AS T2 ON T1.AirportCode  =  T2.DestAirport OR T1.AirportCode  =  T2.SourceA
irport GROUP BY T1.AirportCode ORDER BY count(*) LIMIT 1"
      },
      {
        "question_index": 47,
        "db_id": "flight_2",
        "question": "Give the code of the airport with the least flights.",
        "ori_gold": "SELECT T1.AirportCode FROM AIRPORTS AS T1 JOIN FLIGHTS AS T
2 ON T1.AirportCode  =  T2.DestAirport OR T1.AirportCode  =  T2.SourceAirport GR
OUP BY T1.AirportCode ORDER BY count(*) LIMIT 1",
        "revised_gold": "SELECT T1.AirportCode FROM AIRPORTS AS T1 LEFT JOIN FLI
GHTS AS T2 ON T1.AirportCode  =  T2.DestAirport OR T1.AirportCode  =  T2.SourceA
irport GROUP BY T1.AirportCode ORDER BY count(*) LIMIT 1"
      },
      {
        "question_index": 52,
        "db_id": "flight_2",
        "question": "Find all airlines that have flights from both airports 'APG
' and 'CVO'.",
        "ori_gold": "SELECT T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON
 T1.uid  =  T2.Airline WHERE T2.SourceAirport  =  \"APG\" INTERSECT SELECT T1.Ai
rline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON T1.uid  =  T2.Airline WHERE T2.S
ourceAirport  =  \"CVO\"",
        "revised_gold": "SELECT T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T
2 ON T1.uid  =  T2.Airline WHERE T2.SourceAirport  =  \" APG\" INTERSECT SELECT
T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON T1.uid  =  T2.Airline WHERE
 T2.SourceAirport  =  \" CVO\""
      },
      {
        "question_index": 53,
        "db_id": "flight_2",
        "question": "Which airlines have departing flights from both APG and CVO
 airports?",
        "ori_gold": "SELECT T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON
 T1.uid  =  T2.Airline WHERE T2.SourceAirport  =  \"APG\" INTERSECT SELECT T1.Ai
rline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON T1.uid  =  T2.Airline WHERE T2.S
ourceAirport  =  \"CVO\"",
        "revised_gold": "SELECT T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T
2 ON T1.uid  =  T2.Airline WHERE T2.SourceAirport  =  \" APG\" INTERSECT SELECT
T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON T1.uid  =  T2.Airline WHERE
 T2.SourceAirport  =  \" CVO\""
      },
      {
        "question_index": 54,
        "db_id": "flight_2",
        "question": "Find all airlines that have flights from airport 'CVO' but
not from 'APG'.",
        "ori_gold": "SELECT T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON
 T1.uid  =  T2.Airline WHERE T2.SourceAirport  =  \"CVO\" EXCEPT SELECT T1.Airli
ne FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON T1.uid  =  T2.Airline WHERE T2.Sour
ceAirport  =  \"APG\"",
        "revised_gold": "SELECT T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T
2 ON T1.uid  =  T2.Airline WHERE T2.SourceAirport  =  \" CVO\" EXCEPT SELECT T1.
Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON T1.uid  =  T2.Airline WHERE T2
.SourceAirport  =  \" APG\""
      },
      {
        "question_index": 55,
        "db_id": "flight_2",
        "question": "Which airlines have departures from CVO but not from APG ai
rports?",
        "ori_gold": "SELECT T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON
 T1.uid  =  T2.Airline WHERE T2.SourceAirport  =  \"CVO\" EXCEPT SELECT T1.Airli
ne FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON T1.uid  =  T2.Airline WHERE T2.Sour
ceAirport  =  \"APG\"",
        "revised_gold": "SELECT T1.Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T
2 ON T1.uid  =  T2.Airline WHERE T2.SourceAirport  =  \" CVO\" EXCEPT SELECT T1.
Airline FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON T1.uid  =  T2.Airline WHERE T2
.SourceAirport  =  \" APG\""
      },
      {
        "question_index": 72,
        "db_id": "battle_death",
        "question": "What is the ship id and name that caused most total injurie
s?",
        "ori_gold": "SELECT T2.id ,  T2.name FROM death AS T1 JOIN ship AS t2 ON
 T1.caused_by_ship_id  =  T2.id GROUP BY T2.id ORDER BY count(*) DESC LIMIT 1",
        "revised_gold": "SELECT T2.id ,  T2.name FROM death AS T1 JOIN ship AS t
2 ON T1.caused_by_ship_id  =  T2.id GROUP BY T2.id ORDER BY sum(T1.injured) DESC
 LIMIT 1"
      },
      {
        "question_index": 84,
        "db_id": "student_transcripts_tracking",
        "question": "Which student has enrolled for the most times in any progra
m? List the id, first name, middle name, last name, the number of enrollments an
d student id.",
        "ori_gold": "SELECT T1.student_id ,  T1.first_name ,  T1.middle_name ,
T1.last_name ,  count(*) ,  T1.student_id FROM Students AS T1 JOIN Student_Enrol
ment AS T2 ON T1.student_id  =  T2.student_id GROUP BY T1.student_id ORDER BY co
unt(*) DESC LIMIT 1",
        "revised_gold": "SELECT T1.student_id ,  T1.first_name ,  T1.middle_name
 ,  T1.last_name ,  count(*) FROM Students AS T1 JOIN Student_Enrolment AS T2 ON
 T1.student_id  =  T2.student_id GROUP BY T1.student_id ORDER BY count(*) DESC L
IMIT 1"
      },
      {
        "question_index": 85,
        "db_id": "student_transcripts_tracking",
        "question": "What is the first, middle, and last name, along with the id
 and number of enrollments, for the student who enrolled the most in any program
?",
        "ori_gold": "SELECT T1.student_id ,  T1.first_name ,  T1.middle_name ,
T1.last_name ,  count(*) ,  T1.student_id FROM Students AS T1 JOIN Student_Enrol
ment AS T2 ON T1.student_id  =  T2.student_id GROUP BY T1.student_id ORDER BY co
unt(*) DESC LIMIT 1",
        "revised_gold": "SELECT T1.first_name ,  T1.middle_name ,  T1.last_name
, T1.student_id,count(*) FROM Students AS T1 JOIN Student_Enrolment AS T2 ON T1.
student_id  =  T2.student_id GROUP BY T1.student_id ORDER BY count(*) DESC LIMIT
 1"
      },
      {
        "question_index": 86,
        "db_id": "student_transcripts_tracking",
        "question": "What's the name of the course with most number of enrollmen
ts?",
        "ori_gold": "SELECT  T1.course_name FROM Courses AS T1 JOIN Student_Enro
lment_Courses AS T2 ON T1.course_id  =  T2.course_id GROUP BY T1.course_name ORD
ER BY count(*) DESC LIMIT 1",
        "revised_gold": "SELECT  T1.course_name FROM Courses AS T1 JOIN Student_
Enrolment_Courses AS T2 ON T1.course_id  =  T2.course_id GROUP BY T1.course_id O
```

```
RDER BY count(*) DESC LIMIT 1"
    },
    {
      "question_index": 87,
      "db_id": "student_transcripts_tracking",
      "question": "What is the name of the course with the most students enrol
led?",
      "ori_gold": "SELECT  T1.course_name FROM Courses AS T1 JOIN Student_Enro
lment_Courses AS T2 ON T1.course_id  =  T2.course_id GROUP BY T1.course_name ORD
ER BY count(*) DESC LIMIT 1",
      "revised_gold": "SELECT  T1.course_name FROM Courses AS T1 JOIN Student_
Enrolment_Courses AS T2 ON T1.course_id  =  T2.course_id GROUP BY T1.course_id O
RDER BY count(*) DESC LIMIT 1"
    },
    {
      "question_index": 94,
      "db_id": "student_transcripts_tracking",
      "question": "Find the semester when both Master students and Bachelor st
udents got enrolled in.",
      "ori_gold": "SELECT DISTINCT T2.semester_id FROM Degree_Programs AS T1 J
OIN Student_Enrolment AS T2 ON T1.degree_program_id  =  T2.degree_program_id WHE
RE degree_summary_name  =  'Master' INTERSECT SELECT DISTINCT T2.semester_id FRO
M Degree_Programs AS T1 JOIN Student_Enrolment AS T2 ON T1.degree_program_id  =
 T2.degree_program_id WHERE degree_summary_name  =  'Bachelor'",
      "revised_gold": "SELECT DISTINCT T3.semester_name FROM Degree_Programs A
S T1 JOIN Student_Enrolment AS T2 ON T1.degree_program_id = T2.degree_program_id
 JOIN Semesters AS T3 ON T2.semester_id = T3.semester_id WHERE degree_summary_na
me = 'Master' INTERSECT SELECT DISTINCT T3.semester_name FROM Degree_Programs AS
 T1 JOIN Student_Enrolment AS T2 ON T1.degree_program_id = T2.degree_program_id
JOIN Semesters AS T3 ON T2.semester_id = T3.semester_id WHERE degree_summary_nam
e = 'Bachelor'"
    },
    {
      "question_index": 112,
      "db_id": "world_1",
      "question": "What are the countries where either English or Dutch is the
 official language ?",
      "ori_gold": "select t1.name from country as t1 join countrylanguage as t
```

2 on t1.code = t2.countrycode where t2.language = \"english\" and isofficial
  = \"t\" union select t1.name from country as t1 join countrylanguage as t2 on
 t1.code = t2.countrycode where t2.language = \"dutch\" and isofficial = \
"t\"",
        "revised_gold": "select t1.name from country as t1 join countrylanguage
as t2 on t1.code = t2.countrycode where t2.language = \"english\" and isoffi
cial = \"t\" union select t1.name from country as t1 join countrylanguage as t
2 on t1.code = t2.countrycode where t2.language = \"Dutch\" and isofficial
= \"T\""
      },
      {
        "question_index": 113,
        "db_id": "world_1",
        "question": "Which countries have either English or Dutch as an official
 language?",
        "ori_gold": "SELECT * FROM country AS T1 JOIN countrylanguage AS T2 ON T
1.Code = T2.CountryCode WHERE T2.Language = \"English\" AND IsOfficial = \
"T\" UNION SELECT * FROM country AS T1 JOIN countrylanguage AS T2 ON T1.Code =
 T2.CountryCode WHERE T2.Language = \"Dutch\" AND IsOfficial = \"T\"",
        "revised_gold": "SELECT T1.name FROM country AS T1 JOIN countrylanguage
AS T2 ON T1.Code = T2.CountryCode WHERE T2.Language = \"English\" AND IsOffi
cial = \"T\" UNION SELECT T1.name FROM country AS T1 JOIN countrylanguage AS T
2 ON T1.Code = T2.CountryCode WHERE T2.Language = \"Dutch\" AND IsOfficial
= \"T\""
      },
      {
        "question_index": 116,
        "db_id": "world_1",
        "question": "Find the city with the largest population that uses English
.",
        "ori_gold": "SELECT T1.Name , T1.Population FROM city AS T1 JOIN countr
ylanguage AS T2 ON T1.CountryCode = T2.CountryCode WHERE T2.Language = \"Eng
lish\" ORDER BY T1.Population DESC LIMIT 1",
        "revised_gold": "SELECT T1.Name FROM city AS T1 JOIN countrylanguage AS
T2 ON T1.CountryCode = T2.CountryCode WHERE T2.Language = \"English\" ORDER
BY T1.Population DESC LIMIT 1"
      },
      {

```
        "question_index": 117,
        "db_id": "world_1",
        "question": "What is the most populace city that speaks English?",
        "ori_gold": "SELECT T1.Name ,  T1.Population FROM city AS T1 JOIN countr
ylanguage AS T2 ON T1.CountryCode  =  T2.CountryCode WHERE T2.Language  =  \"Eng
lish\" ORDER BY T1.Population DESC LIMIT 1",
        "revised_gold": "SELECT T1.Name FROM city AS T1 JOIN countrylanguage AS
T2 ON T1.CountryCode  =  T2.CountryCode WHERE T2.Language  =  \"English\" ORDER
BY T1.Population DESC LIMIT 1"
      },
      {
        "question_index": 130,
        "db_id": "world_1",
        "question": "What is the total number of countries where Spanish is spok
en by the largest percentage of people?",
        "ori_gold": "SELECT count(*) ,   max(Percentage) FROM countrylanguage WH
ERE LANGUAGE  =  \"Spanish\" GROUP BY CountryCode",
        "revised_gold": "SELECT COUNT(DISTINCT cl1.CountryCode) FROM countrylang
uage AS cl1 WHERE cl1.Language = 'Spanish' AND cl1.Percentage = (SELECT MAX(Perc
entage) FROM countrylanguage AS cl2 WHERE cl2.CountryCode = cl1.CountryCode);"
      },
      {
        "question_index": 131,
        "db_id": "world_1",
        "question": "Count the number of countries for which Spanish is the pred
ominantly spoken language.",
        "ori_gold": "SELECT count(*) ,   max(Percentage) FROM countrylanguage WH
ERE LANGUAGE  =  \"Spanish\" GROUP BY CountryCode",
        "revised_gold": "SELECT COUNT(DISTINCT cl1.CountryCode) FROM countrylang
uage AS cl1 WHERE cl1.Language = 'Spanish' AND cl1.Percentage = (SELECT MAX(Perc
entage) FROM countrylanguage AS cl2 WHERE cl2.CountryCode = cl1.CountryCode);"
      },
      {
        "question_index": 136,
        "db_id": "network_1",
        "question": "What is the name of the high schooler who has the greatest
number of likes?",
        "ori_gold": "SELECT T2.name FROM Likes AS T1 JOIN Highschooler AS T2 ON
```

```
T1.student_id  =  T2.id GROUP BY T1.student_id ORDER BY count(*) DESC LIMIT 1",
        "revised_gold": "SELECT T2.name FROM Likes AS T1 JOIN Highschooler AS T2
 ON T1.liked_id  =  T2.id GROUP BY T1.liked_id ORDER BY count(*) DESC LIMIT 1"
      },
      {
        "question_index": 137,
        "db_id": "network_1",
        "question": "Give the name of the student with the most likes.",
        "ori_gold": "SELECT T2.name FROM Likes AS T1 JOIN Highschooler AS T2 ON
T1.student_id  =  T2.id GROUP BY T1.student_id ORDER BY count(*) DESC LIMIT 1",
        "revised_gold": "SELECT T2.name FROM Likes AS T1 JOIN Highschooler AS T2
 ON T1.liked_id  =  T2.id GROUP BY T1.liked_id ORDER BY count(*) DESC LIMIT 1"
      },
      {
        "question_index": 148,
        "db_id": "dog_kennels",
        "question": "Which owner has paid for the most treatments on his or her
dogs? List the owner id and last name.",
        "ori_gold": "SELECT T1.owner_id ,  T1.last_name FROM Owners AS T1 JOIN D
ogs AS T2 ON T1.owner_id  =  T2.owner_id JOIN Treatments AS T3 ON T2.dog_id  =
T3.dog_id GROUP BY T1.owner_id ORDER BY count(*) DESC LIMIT 1",
        "revised_gold": "SELECT T1.owner_id ,  T1.last_name FROM Owners AS T1 JO
IN Dogs AS T2 ON T1.owner_id  =  T2.owner_id JOIN Treatments AS T3 ON T2.dog_id
 =  T3.dog_id GROUP BY T1.owner_id ORDER BY SUM(T3.cost_of_treatment) DESC LIMIT
 1"
      },
      {
        "question_index": 149,
        "db_id": "dog_kennels",
        "question": "Tell me the owner id and last name of the owner who spent t
he most on treatments of his or her dogs.",
        "ori_gold": "SELECT T1.owner_id ,  T1.last_name FROM Owners AS T1 JOIN D
ogs AS T2 ON T1.owner_id  =  T2.owner_id JOIN Treatments AS T3 ON T2.dog_id  =
T3.dog_id GROUP BY T1.owner_id ORDER BY count(*) DESC LIMIT 1",
        "revised_gold": "SELECT T1.owner_id ,  T1.last_name FROM Owners AS T1 JO
IN Dogs AS T2 ON T1.owner_id  =  T2.owner_id JOIN Treatments AS T3 ON T2.dog_id
 =  T3.dog_id GROUP BY T1.owner_id ORDER BY SUM(T3.cost_of_treatment) DESC LIMIT
 1"
```

```
    },
    {
      "question_index": 154,
      "db_id": "dog_kennels",
      "question": "What are the first name and last name of the professionals
who have done treatment with cost below average?",
      "ori_gold": "SELECT DISTINCT T1.first_name ,  T1.last_name FROM Professi
onals AS T1 JOIN Treatments AS T2 WHERE cost_of_treatment  <  ( SELECT avg(cost_
of_treatment) FROM Treatments )",
      "revised_gold": "SELECT DISTINCT T1.first_name , T1.last_name FROM Profe
ssionals AS T1 JOIN Treatments AS T2 on T1.professional_id = T2.professional_id
WHERE cost_of_treatment < ( SELECT avg(cost_of_treatment) FROM Treatments )"
    },
    {
      "question_index": 155,
      "db_id": "dog_kennels",
      "question": "Which professionals have operated a treatment that costs le
ss than the average? Give me theor first names and last names.",
      "ori_gold": "SELECT DISTINCT T1.first_name ,  T1.last_name FROM Professi
onals AS T1 JOIN Treatments AS T2 WHERE cost_of_treatment  <  ( SELECT avg(cost_
of_treatment) FROM Treatments )",
      "revised_gold": "SELECT DISTINCT T1.first_name , T1.last_name FROM Profe
ssionals AS T1 JOIN Treatments AS T2 on T1.professional_id = T2.professional_id
WHERE cost_of_treatment < ( SELECT avg(cost_of_treatment) FROM Treatments )"
    },
    {
      "question_index": 158,
      "db_id": "dog_kennels",
      "question": "List the last name of the owner owning the youngest dog.",
      "ori_gold": "SELECT T1.last_name FROM Owners AS T1 JOIN Dogs AS T2 ON T1
.owner_id  =  T2.owner_id WHERE T2.age  =  ( SELECT max(age) FROM Dogs )",
      "revised_gold": "SELECT T1.last_name FROM Owners AS T1 JOIN Dogs AS T2 O
N T1.owner_id  =  T2.owner_id WHERE T2.age  =  ( SELECT min(age) FROM Dogs )"
    },
    {
      "question_index": 159,
      "db_id": "dog_kennels",
      "question": "Who owns the youngest dog? Give me his or her last name.",
        "ori_gold": "SELECT T1.last_name FROM Owners AS T1 JOIN Dogs AS T2 ON T1
.owner_id  =  T2.owner_id WHERE T2.age  =  ( SELECT max(age) FROM Dogs )",
        "revised_gold": "SELECT T1.last_name FROM Owners AS T1 JOIN Dogs AS T2 O
N T1.owner_id  =  T2.owner_id WHERE T2.age  =  ( SELECT min(age) FROM Dogs )"
      }
    ]
  },
  "BIRD dataset": {
    "statistics": "For the BIRD dataset, we corrected the gold SQL for 18 questi
ons. The original gold SQL is marked with the field 'SQL', and the revised gold
SQL is marked with the field 'revised_SQL'",
    "details": [
      {
        "question_id": 87,
        "db_id": "california_schools",
        "question": "What is the e-mail address of the administrator of the scho
ol located in the San Bernardino county, District of San Bernardino City Unified
 that opened between 1/1/2009 to 12/31/2010 whose school types are public Interm
ediate/Middle Schools and Unified Scools?",
        "evidence": "Intermediate/Middle Schools refers to SOC = 62; Unified Sch
ool refers to DOC = 54; years between 2009 and 2010 can refer to 'between 1/1/20
09 to 12/31/2010'",
        "SQL": "SELECT T2.AdmEmail1 FROM frpm AS T1 INNER JOIN schools AS T2 ON
T1.CDSCode = T2.CDSCode WHERE T2.County = 'San Bernardino' AND T2.City = 'San Be
rnardino' AND T2.DOC = 54 AND strftime('%Y', T2.OpenDate) BETWEEN '2009' AND '20
10' AND T2.SOC = 62",
        "revised_SQL": "SELECT CAST(SUM(T1.gender = 'M') AS REAL) * 100 / COUNT(
T1.client_id) FROM client AS T1 INNER JOIN district AS T2 ON T1.district_id = T2
.district_id WHERE T2.A3 = 'south Bohemia' GROUP BY CAST(T2.A4 AS REAL) ORDER BY
 CAST(T2.A4 AS REAL) DESC LIMIT 1",
        "difficulty": "challenging"
      },
      {
        "question_id": 212,
        "db_id": "toxicology",
        "question": "Which element is the least numerous in non-carcinogenic mol
ecules?",
        "evidence": "label = '-' means molecules are non-carcinogenic; element =
```

'cl' means Chlorine; element = 'c' means Carbon; element = 'h' means Hydrogen;
element = 'o' means Oxygen, element = 's' means Sulfur; element = 'n' means Nitr
ogen, element = 'p' means Phosphorus, element = 'na' means Sodium, element = 'br
' means Bromine, element = 'f' means Fluorine; element = 'i' means Iodine; eleme
nt = 'sn' means Tin; element = 'pb' means Lead; element = 'te' means Tellurium;
element = 'ca' means Calcium",
    "SQL": "SELECT T.element FROM (SELECT T1.element, COUNT(DISTINCT T1.mol
ecule_id) FROM atom AS T1 INNER JOIN molecule AS T2 ON T1.molecule_id = T2.molec
ule_id WHERE T2.label = '-' GROUP BY T1.element ORDER BY COUNT(DISTINCT T1.molec
ule_id) ASC LIMIT 4) t",
    "revised_SQL": "SELECT T.element FROM (SELECT T1.element, COUNT(DISTINC
T T1.molecule_id) FROM atom AS T1 INNER JOIN molecule AS T2 ON T1.molecule_id =
T2.molecule_id WHERE T2.label = '-' GROUP BY T1.element ORDER BY COUNT(DISTINCT
T1.molecule_id) ASC LIMIT 1)",
    "difficulty": "challenging"
  },
  {
    "question_id": 281,
    "db_id": "toxicology",
    "question": "Tally the toxicology element of the 4th atom of each molecu
le that was carcinogenic.",
    "evidence": "label = '+' means molecules are carcinogenic; 4th atom of e
ach molecule refers to substr(atom_id, 7, 1) = 4; element = 'cl' means Chlorine;
 element = 'c' means Carbon; element = 'h' means Hydrogen; element = 'o' means O
xygen, element = 's' means Sulfur; element = 'n' means Nitrogen, element = 'p' m
eans Phosphorus, element = 'na' means Sodium, element = 'br' means Bromine, elem
ent = 'f' means Fluorine; element = 'i' means Iodine; element = 'sn' means Tin;
element = 'pb' means Lead; element = 'te' means Tellurium; element = 'ca' means
Calcium",
    "SQL": "SELECT DISTINCT T1.element FROM atom AS T1 INNER JOIN molecule A
S T2 ON T1.molecule_id = T2.molecule_id WHERE T2.label = '+' AND SUBSTR(T1.atom_
id, -1) = '4' AND LENGTH(T1.atom_id) = 7",
    "revised_SQL": "SELECT DISTINCT T1.element FROM atom AS T1 INNER JOIN mo
lecule AS T2 ON T1.molecule_id = T2.molecule_id WHERE T2.label = '+' AND SUBSTR(
T1.atom_id, 7, 1) = '4'",
    "difficulty": "challenging"
  },
  {

```
    "question_id": 282,
    "db_id": "toxicology",
    "question": "What is the ratio of Hydrogen elements in molecule ID TR006
? Please indicate its label.",
    "evidence": "hydrogen refers to element = 'h'; ratio = DIVIDE(SUM(elemen
t = 'h'), count(element)) where molecule_id = 'TR006' ; label = '+' mean molecul
es are carcinogenic; label = '-' means molecules are non-carcinogenic",
    "SQL": "SELECT CAST(COUNT(CASE WHEN T.element = 'h' THEN T.atom_id ELSE
NULL END) AS REAL) / COUNT(T.atom_id) FROM ( SELECT DISTINCT T1.atom_id, T1.elem
ent, T1.molecule_id, T2.label FROM atom AS T1 INNER JOIN molecule AS T2 ON T1.mo
lecule_id = T2.molecule_id WHERE T2.molecule_id = 'TR006' ) AS T UNION ALL SELEC
T DISTINCT T3.label FROM ( SELECT DISTINCT T1.atom_id, T1.element, T1.molecule_i
d, T2.label FROM atom AS T1 INNER JOIN molecule AS T2 ON T1.molecule_id = T2.mol
ecule_id WHERE T2.molecule_id = 'TR006' ) AS T3",
    "revised_SQL": "SELECT DISTINCT T.label, CAST(COUNT(CASE WHEN T.element
= 'h' THEN T.atom_id ELSE NULL END) AS REAL) / COUNT(T.atom_id) FROM ( SELECT DI
STINCT T1.atom_id, T1.element, T1.molecule_id, T2.label FROM atom AS T1 INNER JO
IN molecule AS T2 ON T1.molecule_id = T2.molecule_id WHERE T2.molecule_id = 'TR0
06' ) AS T",
    "difficulty": "challenging"
  },
  {
    "question_id": 330,
    "db_id": "toxicology",
    "question": "Calculate the percentage of carcinogenic molecules with tri
ple bonded Hidrogen atoms.",
    "evidence": "hydrogen refers to element = 'h'; label = '+' mean molecule
s are carcinogenic; triple bond refers to bond_type = '#'; percentage = DIVIDE(S
UM(label = '+'), COUNT(molecule_id)) * 100.0 where element = 'h' AND bond_type =
 '#';",
    "SQL": "SELECT CAST(SUM(CASE WHEN T1.label = '+' THEN 1 ELSE 0 END) AS R
EAL) * 100 / COUNT(DISTINCT T1.molecule_id) FROM molecule AS T1 INNER JOIN atom
AS T2 ON T1.molecule_id = T2.molecule_id INNER JOIN bond AS T3 ON T1.molecule_id
 = T3.molecule_id WHERE T3.bond_type = '#' AND T2.element = 'h'",
    "revised_SQL": "SELECT CAST(COUNT(DISTINCT CASE WHEN T1.label = '+' THEN
 T1.molecule_id ELSE NULL END) AS REAL) * 100 / COUNT(DISTINCT T1.molecule_id) F
ROM molecule AS T1 INNER JOIN atom AS T2 ON T1.molecule_id = T2.molecule_id INNE
R JOIN bond AS T3 ON T1.molecule_id = T3.molecule_id WHERE T3.bond_type = '#' AN
```

D T2.element = 'h'",
        "difficulty": "challenging"
      },
      {
        "question_id": 416,
        "db_id": "card_games",
        "question": "What percentage of cards without power are in French?",
        "evidence": "in French refers to language = 'French'; cards without powe
r refers to power IS NULL OR power = '*'; \npercentage = DIVIDE(COUNT(language =
 'French' and power is NULL or power = '*'), COUNT(power is NULL or power = '*'
))*100\n",
        "SQL": "SELECT CAST(SUM(CASE WHEN T2.language = 'French' THEN 1 ELSE 0 E
ND) AS REAL) * 100 / COUNT(T1.id) FROM cards AS T1 INNER JOIN foreign_data AS T2
 ON T1.uuid = T2.uuid WHERE T1.power IS NULL OR T1.power LIKE '%*%'",
        "revised_SQL": "SELECT CAST(SUM(CASE WHEN T2.language = 'French' THEN 1
ELSE 0 END) AS REAL) * 100 / COUNT(T1.id) FROM cards AS T1 INNER JOIN foreign_da
ta AS T2 ON T1.uuid = T2.uuid WHERE T1.power IS NULL OR T1.power = '*'",
        "difficulty": "challenging"
      },
      {
        "question_id": 487,
        "db_id": "card_games",
        "question": "What is the percentage of incredibly powerful cards in the
set Coldsnap?",
        "evidence": "card set Coldsnap refers to name = 'Coldsnap'; foil is incr
edibly powerful refers to cardKingdomFoilId = cardKingdomId AND cardKingdomId is
 not null; the percentage of incredibly powerful cards in the set refers to DIVI
DE(SUM(incredibly powerful), SUM(name = 'Coldsnap'))*100\n\n",
        "SQL": "SELECT CAST(SUM(CASE WHEN T1.cardKingdomFoilId IS NOT NULL AND T
1.cardKingdomId IS NOT NULL THEN 1 ELSE 0 END) AS REAL) * 100 / COUNT(T1.id) FRO
M cards AS T1 INNER JOIN sets AS T2 ON T2.code = T1.setCode WHERE T2.name = 'Col
dsnap'",
        "revised_SQL": "SELECT CAST(SUM(CASE WHEN T1.cardKingdomId = T1.cardKing
domFoilId AND T1.cardKingdomFoilId IS NOT NULL AND T1.cardKingdomId IS NOT NULL
THEN 1 ELSE 0 END) AS REAL) * 100 / COUNT(T1.id) FROM cards AS T1 INNER JOIN set
s AS T2 ON T2.code = T1.setCode WHERE T2.name = 'Coldsnap'",
        "difficulty": "challenging"
      },

```
    {
      "question_id": 494,
      "db_id": "card_games",
      "question": "For all cards illustrated by Jim Pavelec. and describe the
text of the ruling of these cards. Do these cards have missing or degraded prope
rties and values.",
      "evidence": "all cards illustrated by Jim Pavelec refers to artist = 'Ji
m Pavelec'; the text of the ruling refers to text; cards have missing or degrade
d properties and values if hasContentWarning = 1 else it doesn't have;",
      "SQL": "SELECT T2.text , CASE WHEN T1.hasContentWarning = 1 THEN 'YES' E
LSE 'NO' END FROM cards AS T1 INNER JOIN rulings AS T2 ON T2.uuid = T1.uuid WHER
E T1.artist = 'Jim Pavelec'",
      "revised_SQL": "SELECT T2.text , T1.hasContentWarning FROM cards AS T1 I
NNER JOIN rulings AS T2 ON T2.uuid = T1.uuid WHERE T1.artist = 'Jim Pavelec'",
      "difficulty": "challenging"
    },
    {
      "question_id": 523,
      "db_id": "card_games",
      "question": "What is the annual average number of sets that were release
d between 1/1/2012 to 12/31/2015? Indicate the common langugage of the card.",
      "evidence": "AVG(id); releaseDate BETWEEN 1/1/2012 AND 12/31/2015; the c
ommon language refers to MAX(COUNT(language))",
      "SQL": "SELECT (CAST(SUM(T1.id) AS REAL) / COUNT(T1.id)) / 4, T2.languag
e FROM sets AS T1 INNER JOIN set_translations AS T2 ON T1.id = T2.id WHERE T1.re
leaseDate BETWEEN '2012-01-01' AND '2015-12-31' GROUP BY T1.releaseDate ORDER BY
 COUNT(T2.language) DESC LIMIT 1",
      "revised_SQL": "SELECT (CAST(COUNT(DISTINCT T1.id) AS REAL) / COUNT(DIST
INCT T1.id)) / 4, T2.language FROM sets AS T1 INNER JOIN set_translations AS T2
ON T1.id = T2.id WHERE T1.releaseDate BETWEEN '2012-01-01' AND '2015-12-31' GROU
P BY T1.releaseDate ORDER BY COUNT(T2.language) DESC LIMIT 1",
      "difficulty": "challenging"
    },
    {
      "question_id": 528,
      "db_id": "card_games",
      "question": "List the names of all the cards in the set Hour of Devastat
ion and find the formats in which these cards are legal.",
        "evidence": "the set Hour of Devastation refers to set.name = 'Hour of D
evastation'; names of all the cards in the set refers to cards.name; legal cards
 refers to status = 'legal'; the formats refers to format",
        "SQL": "SELECT DISTINCT T2.name , CASE WHEN T1.status = 'Legal' THEN T1.
format ELSE NULL END FROM legalities AS T1 INNER JOIN cards AS T2 ON T2.uuid = T
1.uuid WHERE T2.setCode IN ( SELECT code FROM sets WHERE name = 'Hour of Devasta
tion' )",
        "revised_SQL": "SELECT DISTINCT T2.name , T1.format FROM legalities AS T
1 INNER JOIN cards AS T2 ON T2.uuid = T1.uuid WHERE T1.status = 'Legal' AND T2.s
etCode IN ( SELECT code FROM sets WHERE name = 'Hour of Devastation' )",
        "difficulty": "challenging"
      },
      {
        "question_id": 639,
        "db_id": "codebase_community",
        "question": "Based on posts posted by Community, calculate the percentag
e of posts that use the R language.",
        "evidence": "DIVIDE(COUNT(PostId WHERE TagName = R language)), (COUNT(Po
stId WHERE DisplayName = 'Community')) as percentage; R language refers to tagna
me = 'r'",
        "SQL": "SELECT CAST(SUM(IIF(T3.TagName = 'r', 1, 0)) AS REAL) * 100 / CO
UNT(T1.Id) FROM users AS T1 INNER JOIN postHistory AS T2 ON T1.Id = T2.UserId IN
NER JOIN tags AS T3 ON T3.ExcerptPostId = T2.PostId WHERE T1.DisplayName = 'Comm
unity'",
        "revised_SQL": "SELECT CAST(COUNT(DISTINCT IIF(T3.TagName = 'r', T1.Id,
NULL)) AS REAL) * 100 / COUNT(DISTINCT T1.Id) FROM users AS T1 INNER JOIN postHi
story AS T2 ON T1.Id = T2.UserId INNER JOIN tags AS T3 ON T3.ExcerptPostId = T2.
PostId WHERE T1.DisplayName = 'Community'",
        "difficulty": "challenging"
      },
      {
        "question_id": 773,
        "db_id": "superhero",
        "question": "Which superhero has the same eyes, hair and skin colour? In
dicate the publisher of the superhero.",
        "evidence": "which superhero refers to superhero_name; the same eyes, ha
ir and skin colour refers to hair_colour_id = skin_colour_id AND hair_colour_id
= eye_colour_id; publisher refers to publisher_name;",
```

"SQL": "SELECT T1.superhero_name, T2.publisher_name FROM superhero AS T1
INNER JOIN publisher AS T2 ON T1.publisher_id = T2.id WHERE T1.eye_colour_id =
T1.hair_colour_id AND T1.eye_colour_id = T1.skin_colour_id",
"revised_SQL": "SELECT T1.superhero_name, T2.publisher_name FROM superhe
ro AS T1 INNER JOIN publisher AS T2 ON T1.publisher_id = T2.id WHERE T1.hair_col
our_id = T1.skin_colour_id AND T1.hair_colour_id = T1.eye_colour_id",
"difficulty": "challenging"
},
{
"question_id": 896,
"db_id": "formula_1",
"question": "Calculate the percentage whereby Hamilton was not at the 1s
t track of the the f1 circuit since 2010.",
"evidence": "DIVIDE(COUNT(raceId) where surname = 'Hamilton', year >= 20
10 and position>1), (COUNT(raceId) where surname = 'Hamilton', year >= 2010) as
percentage;",
"SQL": "SELECT CAST(COUNT(CASE WHEN T2.position <> 1 THEN T2.position EN
D) AS REAL) * 100 / COUNT(T2.driverStandingsId) FROM races AS T1 INNER JOIN driv
erStandings AS T2 ON T2.raceId = T1.raceId INNER JOIN drivers AS T3 ON T3.driver
Id = T2.driverId WHERE T3.surname = 'Hamilton' AND T1.year >= 2010",
"revised_SQL": "SELECT CAST(COUNT(DISTINCT CASE WHEN T2.position <> 1 TH
EN T2.driverStandingsId ELSE NULL END) AS REAL) * 100 / COUNT(DISTINCT T2.driver
StandingsId) FROM races AS T1 INNER JOIN driverStandings AS T2 ON T2.raceId = T1
.raceId INNER JOIN drivers AS T3 ON T3.driverId = T2.driverId WHERE T3.surname =
'Hamilton' AND T1.year >= 2010",
"difficulty": "challenging"
},
{
"question_id": 1242,
"db_id": "thrombosis_prediction",
"question": "For laboratory examinations take in 1984, list all patients
below 50 years old with normal platelet level.",
"evidence": "laboratory examinations take in 1984 refers to Date like '1
984%'; below 50 years old = SUBTRACT(year(current_timestamp), year(Birthday)) <
50; normal platelet level refers to PLT between 100 and 400;",
"SQL": "SELECT DISTINCT T1.ID FROM Patient AS T1 INNER JOIN Laboratory A
S T2 ON T1.ID = T2.ID WHERE T2.PLT BETWEEN 100 AND 400 AND STRFTIME('%Y', T2.Dat
e) - STRFTIME('%Y', T1.Birthday) < 50 AND STRFTIME('%Y', T2.Date) = '1984'",

```
        "revised_SQL": "SELECT DISTINCT T1.ID FROM Patient AS T1 INNER JOIN Labo
ratory AS T2 ON T1.ID = T2.ID WHERE T2.PLT BETWEEN 100 AND 400 AND STRFTIME('%Y'
, CURRENT_TIMESTAMP) - STRFTIME('%Y', T1.Birthday) < 50 AND STRFTIME('%Y', T2.Da
te) = '1984'",
        "difficulty": "challenging"
      },
      {
        "question_id": 1243,
        "db_id": "thrombosis_prediction",
        "question": "For all patients who are older than 55 years old, what is t
he percentage of female who has abnormal prothrombin time (PT)?",
        "evidence": "older than 55 years old = SUBTRACT(year(current_timestamp),
 year(Birthday)) > 55; percentage = MULTIPLY(DIVIDE(SUM(PT > = 14 AND SEX = 'F')
, SUM(PT > = 14)), 1.0); female refers to Sex = 'F'; abnormal prothrombin time (
PT) refers to PT > = 14;",
        "SQL": "SELECT CAST(SUM(CASE WHEN T2.PT >= 14 AND T1.SEX = 'F' THEN 1 EL
SE 0 END) AS REAL) * 100 / COUNT(*) FROM Patient AS T1 INNER JOIN Laboratory AS
T2 ON T1.ID = T2.ID WHERE STRFTIME('%Y', CURRENT_TIMESTAMP) - STRFTIME('%Y', T1.
Birthday) > 55",
        "revised_SQL": "SELECT CAST(SUM(CASE WHEN T1.SEX = 'F' THEN 1 ELSE 0 END
) AS REAL) * 100 / COUNT(*) FROM Patient AS T1 INNER JOIN Laboratory AS T2 ON T1
.ID = T2.ID WHERE T2.PT >= 14 AND (STRFTIME('%Y', CURRENT_TIMESTAMP) - STRFTIME(
'%Y', T1.Birthday) > 55)",
        "difficulty": "challenging"
      },
      {
        "question_id": 1257,
        "db_id": "thrombosis_prediction",
        "question": "Among the patients whose C-reactive protein level is abnorm
al, how many of them aren't 18 yet?",
        "evidence": "C-reactive protein level is abnormal refers to CRP NOT IN('
+-', '-') AND CRP > = 1.0; aren't 18 refers = CRP > = 1.0 AND SUBTRACT((YEAR(CUR
DATE()), YEAR(Birthday))) < 18; Should compute the number of distinct ones.",
        "SQL": "SELECT COUNT(DISTINCT T1.ID) FROM Patient AS T1 INNER JOIN Labor
atory AS T2 ON T1.ID = T2.ID WHERE (T2.CRP != '-' AND T2.CRP != '+-') AND T2.CRP
 >= 1.0 AND STRFTIME('%Y', Date('now')) - STRFTIME('%Y', T1.Birthday) < '18'",
        "revised_SQL": "SELECT COUNT(DISTINCT T1.ID) FROM Patient AS T1 INNER JO
IN Laboratory AS T2 ON T1.ID = T2.ID WHERE (T2.CRP != '-' AND T2.CRP != '+-') AN
```

```
D T2.CRP >= 1.0 AND STRFTIME('%Y', Date('now')) - STRFTIME('%Y', T1.Birthday) <
18",
        "difficulty": "challenging"
      },
      {
        "question_id": 1359,
        "db_id": "student_club",
        "question": "How many times was the budget in Advertisement for \"Yearly
 Kickoff\" meeting more than \"October Meeting\"?",
        "evidence": "DIVIDE(SUM(amount where category = 'Advertisement' and even
t_name = 'Yearly Kickoff'), SUM(amount event_name = 'October Meeting' and catego
ry = 'Advertisement'))",
        "SQL": "SELECT CAST(SUM(CASE WHEN T2.event_name = 'Yearly Kickoff' THEN
T1.amount ELSE 0 END) AS REAL) / SUM(CASE WHEN T2.event_name = 'October Meeting'
 THEN T1.amount ELSE 0 END) FROM budget AS T1 INNER JOIN event AS T2 ON T1.link_
to_event = T2.event_id WHERE T1.category = 'Advertisement' AND T2.type = 'Meetin
g'",
        "revised_SQL": "SELECT CAST(SUM(CASE WHEN T2.event_name = 'Yearly Kickof
f' THEN T1.amount ELSE 0 END) AS REAL) / SUM(CASE WHEN T2.event_name = 'October
Meeting' THEN T1.amount ELSE 0 END) FROM budget AS T1 INNER JOIN event AS T2 ON
T1.link_to_event = T2.event_id WHERE T1.category = 'Advertisement'",
        "difficulty": "challenging"
      },
      {
        "question_id": 1482,
        "db_id": "debit_card_specializing",
        "question": "Which of the three segments—SME, LAM and KAM—has the bigges
t and lowest percentage increases in consumption paid in EUR between 2012 and 20
13?",
        "evidence": "Increase or Decrease = consumption for 2013 - consumption f
or 2012; Percentage of Increase = (Increase or Decrease / consumption for 2013)
* 100%; Between 2012 And 2013 can be represented by Between 201201 And 201312; F
irst 4 strings of Date represents the year.",
        "SQL": "SELECT CAST((SUM(IIF(T1.Segment = 'SME' AND T2.Date LIKE '2013%'
, T2.Consumption, 0)) - SUM(IIF(T1.Segment = 'SME' AND T2.Date LIKE '2012%', T2.
Consumption, 0))) AS FLOAT) * 100 / SUM(IIF(T1.Segment = 'SME' AND T2.Date LIKE
'2012%', T2.Consumption, 0)), CAST(SUM(IIF(T1.Segment = 'LAM' AND T2.Date LIKE '
2013%', T2.Consumption, 0)) - SUM(IIF(T1.Segment = 'LAM' AND T2.Date LIKE '2012%
```

', T2.Consumption, 0)) AS FLOAT) * 100 / SUM(IIF(T1.Segment = 'LAM' AND T2.Date
LIKE '2012%', T2.Consumption, 0)) , CAST(SUM(IIF(T1.Segment = 'KAM' AND T2.Date
LIKE '2013%', T2.Consumption, 0)) - SUM(IIF(T1.Segment = 'KAM' AND T2.Date LIKE
'2012%', T2.Consumption, 0)) AS FLOAT) * 100 / SUM(IIF(T1.Segment = 'KAM' AND T2
.Date LIKE '2012%', T2.Consumption, 0)) FROM customers AS T1 INNER JOIN yearmont
h AS T2 ON T1.CustomerID = T2.CustomerID",
        "revised_SQL": " WITH sub1 AS (SELECT T1.segment AS n1, (SUM(CASE WHEN T
2.date LIKE '2013%' THEN T2.consumption ELSE 0 END) - SUM(CASE WHEN T2.date LIKE
 '2012%' THEN T2.consumption ELSE 0 END)) * 100 / SUM(CASE WHEN T2.date LIKE '20
12%' THEN T2.consumption ELSE 0 END) AS percentage_increase FROM customers AS T1
 INNER JOIN yearmonth AS T2 ON T1.customerid = T2.customerid WHERE T1.segment IN
 ('SME', 'LAM', 'KAM') GROUP BY T1.segment ORDER BY percentage_increase DESC LIM
IT 1), sub2 AS (SELECT T1.segment AS n2, (SUM(CASE WHEN T2.date LIKE '2013%' THE
N T2.consumption ELSE 0 END) - SUM(CASE WHEN T2.date LIKE '2012%' THEN T2.consum
ption ELSE 0 END)) * 100 / SUM(CASE WHEN T2.date LIKE '2012%' THEN T2.consumptio
n ELSE 0 END) AS percentage_increase FROM customers AS T1 INNER JOIN yearmonth A
S T2 ON T1.customerid = T2.customerid WHERE T1.segment IN ('SME', 'LAM', 'KAM')
GROUP BY T1.segment ORDER BY percentage_increase ASC LIMIT 1) SELECT sub1.n1, su
b1.percentage_increase FROM sub1 UNION SELECT sub2.n2, sub2.percentage_increase
FROM sub2 ; WITH sub1 AS (SELECT T1.segment AS n1, (SUM(CASE WHEN T2.date LIKE '
2013%' THEN T2.consumption ELSE 0 END) - SUM(CASE WHEN T2.date LIKE '2012%' THEN
 T2.consumption ELSE 0 END)) * 100 / SUM(CASE WHEN T2.date LIKE '2012%' THEN T2.
consumption ELSE 0 END) AS percentage_increase FROM customers AS T1 INNER JOIN y
earmonth AS T2 ON T1.customerid = T2.customerid WHERE T1.segment IN ('SME', 'LAM
', 'KAM') GROUP BY T1.segment ORDER BY percentage_increase DESC LIMIT 1), sub2 A
S (SELECT T1.segment AS n2, (SUM(CASE WHEN T2.date LIKE '2013%' THEN T2.consumpt
ion ELSE 0 END) - SUM(CASE WHEN T2.date LIKE '2012%' THEN T2.consumption ELSE 0
END)) * 100 / SUM(CASE WHEN T2.date LIKE '2012%' THEN T2.consumption ELSE 0 END)
 AS percentage_increase FROM customers AS T1 INNER JOIN yearmonth AS T2 ON T1.cu
stomerid = T2.customerid WHERE T1.segment IN ('SME', 'LAM', 'KAM') GROUP BY T1.s
egment ORDER BY percentage_increase ASC LIMIT 1) SELECT sub1.n1 FROM sub1 UNION
SELECT sub2.n2 FROM sub2",
        "difficulty": "challenging"
      }
    ]
  }
}