Dominique W. Chandra D.
6162101104
Mata Kuliah Rekayasa Data

Tugas 1

1. Jika kolom pokemon_id dihapus maka kolom name dapat digunakan sebagai primary key karena tidak ada nama pokemon yang sama artinya dalam kolom name tidak ada duplikasi sehingga kolom name bisa digunakan sebagai primary key.
   Kode SQL:
   ```
   SELECT pokemon_name, COUNT(*) AS duplicate_name
   FROM pokemon
   GROUP BY pokemon_name
   HAVING COUNT(*) > 1;
   ```
   Hasil:

	pokemon_name	duplicate_name

2. Terdapat 1 kolom pada tabel pokemon yang memiliki nilai kosong yaitu kolom secondary_type. Jika dilihat, kolom secondary_type adalah kolom yang berisi tipe sekunder dari pokemon. Perlu diketahui bahwa tidak semua pokemon memiliki tipe sekunder, sehingga kolom yang kosong pada kolom secondary_type akan diabaikan saja.
   Kode SQL:
   ```
   SELECT COUNT(*) AS empty_count
   FROM pokemon
   WHERE secondary_type = '';
   ```
   Hasil:

	empty_count
▶	499

3. Tabel yang diberikan memang tidak memiliki baris duplikat jika dilihat pada hasil yang dikeluarkan. Namun keunikan terjadi jika kolom pokemon_id dan pokemon_name tidak dimasukkan maka akan muncul keluaran seperti ini. Artinya terdapat 2 pokemon yang memiliki informasi yang sama persis. Jika dilakukan analisis lebih lanjut maka dapat dilihat bahwa 2 pokemon yang memiliki informasi identik ini berasal dari 2 pokemon yang berbeda yaitu cascoon dan silcoon.
   Kode SQL:
   ```
   SELECT
      GROUP_CONCAT(DISTINCT pokemon_name ORDER BY pokemon_name
   SEPARATOR ', ') AS different_names,
      primary_type, secondary_type, first_appereance, generation, category,
      total_base_stats, hp, attack, defense, special_attack, special_defense, speed,
   ```

```
    COUNT(*) AS duplicate_count
FROM pokemon
GROUP BY primary_type, secondary_type, first_appereance, generation, category,
        total_base_stats, hp, attack, defense, special_attack, special_defense, speed
HAVING COUNT(*) > 1;
```

Hasil:

different_names	primary_type	secondary_type	first_appereance	generation	category	total_base_stats	hp	attack	defense	special_attack	special_defense	speed	duplicate_count
cascoon, silcoon	bug		ruby/sapphire	gen 3	regular	205	50	35	55	25	25	15	2

4. Tipe data dari masing-masing kolom:

Nama Kolom	Tipe Data
pokemon_id	numerik (integer)
pokemon_name	Karakter (varchar(255))
primary_type	Karakter (varchar(255))
secondary_type	Karakter (varchar(255))
first_appearance	Karakter (varchar(255))
generation	Karakter (varchar(255))
category	Karakter (varchar(255))
total_base_stats	numerik (integer)
Hp, attack, defence, special_attack, special_defense, speed	numerik (integer)

5. Validasi format dapat dilakukan ketika membuat tabel dengan:

```
CREATE TABLE pokemon (
    pokemon_id INT,
    pokemon_name VARCHAR(255),
    primary_type VARCHAR(255),
    secondary_type VARCHAR(255),
    first_appereance VARCHAR(255),
    generation VARCHAR(255),
    category VARCHAR(255),
    total_base_stats INT,
    hp INT,
    attack INT,
    defense INT,
```

```
    special_attack INT,
    special_defense INT,
    speed INT,
    PRIMARY KEY (pokemon_id, pokemon_name)
);
```

6. Didapat hasil summary untuk setiap tipe data numerik adalah sebagai berikut

Column name	Rata-rata	Standar deviasi	Nilai maksimum	Nilai minimum
total_base_stats	427,68	112.71	720	175
hp	70.18	26.61	255	1
attack	77.52	29.76	181	5
defense	72.50	29.27	230	5
special_attack	70.08	29.64	173	10
special_defense	70.20	26.62	230	20
speed	67.18	28.70	200	5

Jika dilihat dari tabel diatas, rata-rata setiap stats pokemon baik hp, attack, defense, special attack, special defense, dan speed berada di nilai 70. Namun, jika dilihat nilai maksimum pada setiap stats terdapat indikasi ada data pencilan/outlier. Hal yang sama juga berlaku pada total_base_stats, dengan rata-rata di sekitar nilai 427 namun terdapat pokemon yang memiliki nilai maksimum cukup jauh dari rata-rata yaitu 720, sehingga kemungkinan terdapat data pencilan juga.

Kode SQL:

```
SELECT
    'total_base_stats' AS column_name, AVG(total_base_stats) AS average,
STDDEV(total_base_stats) AS std, MAX(total_base_stats) AS max,
MIN(total_base_stats) AS min
FROM pokemon
UNION ALL
SELECT
    'hp', AVG(hp), STDDEV(hp), MAX(hp), MIN(hp)
FROM pokemon
UNION ALL
SELECT
    'attack', AVG(attack), STDDEV(attack), MAX(attack), MIN(attack)
FROM pokemon
UNION ALL
SELECT
```

```
    'defense', AVG(defense), STDDEV(defense), MAX(defense), MIN(defense)
FROM pokemon
UNION ALL
SELECT
    'special_attack', AVG(special_attack), STDDEV(special_attack), MAX(special_attack),
MIN(special_attack)
FROM pokemon
UNION ALL
SELECT
    'special_defense', AVG(special_defense), STDDEV(special_defense),
MAX(special_defense), MIN(special_defense)
FROM pokemon
UNION ALL
SELECT
    'speed', AVG(speed), STDDEV(speed), MAX(speed), MIN(speed)
FROM pokemon;
```

7. Berikut ini adalah tabel yang berisikan frekuensi setiap kategori di kolom primary type:

kategori	frekuensi
grass	103
fire	66
water	134
bug	83
normal	118
poison	42
electric	59
ground	40
fairy	29
fighting	40
psychic	60
rock	58
ghost	35
ice	31
dragon	37
dark	45
steel	36
flying	9

Jika dilihat pada tabel di atas, tipe utama pokemon terbanyak adalah tipe air/water dan tipe utama paling sedikit adalah tipe terbang/flying. Selanjutnya, tabel di bawah ini menunjukkan frekuensi setiap kategori di kolom secondary type:

Kategori	Frekuensi
poison	41
	499
flying	100
ground	35
fairy	35
grass	24
fighting	33
psychic	42
steel	29
ice	17
rock	16
water	20
electric	10
fire	15
dragon	33
dark	24
ghost	30
bug	9
normal	13

Tipe sekunder pokemon terbanyak adalah tipe terbang/flying dan tipe sekunder pokemon paling sedikit adalah tipe serangga/bug. Selain itu, jika diperhatikan pada tabel di atas, dapat dilihat bahwa terdapat kolom kosong dengan frekuensi 499, artinya terdapat sebanyak 499 pokemon yang tidak memiliki tipe sekunder. Selanjutnya, terdapat tabel yang berisi frekuensi dari kolom first appereance:

First appereance	Frekuensi
red/blue	151
gold/silver	100
ruby/sapphire	135
diamond/pearl	107

black/white	156
x/y	72
sun/moon	88
sword/shield	96
scarlet/violet	120

Jika dilihat pada tabel tersebut, kebanyakan pokemon muncul pertama kali di versi Pokemon Black and White yaitu sebanyak 156 pokemon dan hanya 72 pokemon baru yang pertama kali dikenalkan di versi Pokemon X and Y. Terakhir, terdapat tabel yang berisi frekuensi dari kolom category:

Kategori	Frekuensi
regular	902
legendary	70
mythical	23
ultra beast	11
paradox	19

Jika dilihat pada tabel tersebut, terdapat 902 pokemon reguler artinya 88% dari total pokemon (terdapat total 1025 pokemon) diklasifikasikan sebagai pokemon biasa/reguler dan pokemon dengan kategori paling sedikit adalah pokemon ultra beast.
Kode SQL:

```
SELECT 'primary_type' AS column_name, primary_type AS kategori, COUNT(*) AS frekuensi
FROM pokemon GROUP BY primary_type
UNION ALL
SELECT 'secondary_type', secondary_type, COUNT(*)
FROM pokemon GROUP BY secondary_type
UNION ALL
SELECT 'first_appereance', first_appereance, COUNT(*)
FROM pokemon GROUP BY first_appereance
UNION ALL
SELECT 'generation', generation, COUNT(*)
FROM pokemon GROUP BY generation
UNION ALL
SELECT 'category', category, COUNT(*)
FROM pokemon GROUP BY category;
```

8. Tabel yang berisikan informasi Pokemon dengan nama berawalan huruf X:

pokemon_id	pokemon_name	primary_type	secondary_type	first_appereance	generation	category	total_base_stats	hp	attack	defense	special_attack	special_defense	speed
178	xatu	psychic	flying	gold/silver	gen 2	regular	470	65	75	70	95	70	95
716	xerneas	fairy		x/y	gen 6	legendary	680	126	131	95	131	98	99
796	xurkitree	electric		sun/moon	gen 7	ultra beast	570	83	89	71	173	71	83

Kode SQL:

```
SELECT *
FROM pokemon
WHERE pokemon_name LIKE 'X%';
```

9. Tabel yang berisikan informasi mengenai pokemon dengan nama paling panjang:

pokemon_id	pokemon_name	primary_type	secondary_type	first_appereance	generation	category	total_base_stats	hp	attack	defense	special_attack	special_defense	speed
740	crabominable	fighting	ice	sun/moon	gen 7	regular	478	97	132	77	62	67	43
931	squawkabilly	normal	flying	scarlet/violet	gen 9	regular	417	82	96	51	45	51	92
947	brambleghast	grass	ghost	scarlet/violet	gen 9	regular	480	55	115	70	80	70	90
986	brute-bonnet	grass	dark	scarlet/violet	gen 9	paradox	570	111	127	99	79	99	55
987	flutter-mane	ghost	fairy	scarlet/violet	gen 9	paradox	570	55	55	55	135	135	135
988	slither-wing	bug	fighting	scarlet/violet	gen 9	paradox	570	85	135	79	85	105	81
989	sandy-shocks	electric	ground	scarlet/violet	gen 9	paradox	570	85	81	97	121	85	101
993	iron-jugulis	dark	flying	scarlet/violet	gen 9	paradox	570	94	80	86	122	80	108
1005	roaring-moon	dragon	dark	scarlet/violet	gen 9	paradox	590	105	139	71	55	101	119
1006	iron-valiant	fairy	fighting	scarlet/violet	gen 9	paradox	590	74	130	90	120	60	116
1009	walking-wake	water	dragon	scarlet/violet	gen 9	paradox	590	99	83	91	125	83	109
1012	poltchageist	grass	ghost	scarlet/violet	gen 9	regular	308	40	45	45	74	54	50
1020	gouging-fire	fire	dragon	scarlet/violet	gen 9	paradox	590	105	115	121	65	93	91
1022	iron-boulder	rock	psychic	scarlet/violet	gen 9	regular	590	90	120	80	68	108	124

Terdapat 14 Pokemon dengan nama yang panjang

Kode SQL:

```
SELECT *
FROM pokemon
WHERE LENGTH(pokemon_name) = (SELECT MAX(LENGTH(pokemon_name))
FROM pokemon);
```

10. Tipe pertama (primary type) terbanyak adalah

primary_type	total_count
water	134

Kode SQL:

```
SELECT primary_type, COUNT(*) AS total_count
FROM pokemon
GROUP BY primary_type
ORDER BY total_count DESC
LIMIT 1;
```

11. Jika ditulis notasi aljabar relasional nya maka kueri untuk menampilkan seluruh nama dan generasi yang memiliki tipe pertama dan kedua adalah

$$\Pi_{name,generation}(\sigma_{secondary\_type\neq' '}(\text{pokemon}))$$

Kode SQL:

```
SELECT pokemon_name, generation
FROM pokemon
WHERE secondary_type <> '';
```

Hasil yang diberikan: Terdapat 526 Pokemon yang memiliki tipe pertama dan tipe kedua

pokemon_name	generation
bulbasaur	gen 1
ivysaur	gen 1
venusaur	gen 1
charizard	gen 1
butterfree	gen 1
weedle	gen 1
kakuna	gen 1
beedrill	gen 1
pidgey	gen 1
pidgeotto	gen 1

12. Kueri untuk menampilkan ID, nama, dan status individual pokemon yang mempunyai special attack di atas rata-rata adalah

$$avg\_special\_attack \leftarrow_{AVG(special\_attack)} (\text{pokemon})$$

$$\Pi_{ID,name,hp,attack,defense,special\_attack,special\_defense,speed}(\sigma_{special\_attack>avg\_special\_attack}(\text{pokemon}))$$

Kode SQL:

```
SELECT pokemon_id, pokemon_name, hp, attack, defense, special_attack,
special_defense, speed
 FROM pokemon
 WHERE special_attack > (SELECT AVG(special_attack) FROM pokemon);
```

Hasil yang diberikan: Terdapat total 426 Pokemon yang memiliki special attack di atas rata-rata

pokemon_id	pokemon_name	hp	attack	defense	special_attack	special_defense	speed
2	ivysaur	60	62	63	80	80	60
3	venusaur	80	82	83	100	100	80
5	charmeleon	58	64	58	80	65	80
6	charizard	78	84	78	109	85	100
9	blastoise	79	83	100	85	105	78
12	butterfree	60	45	50	90	80	70
26	raichu	60	90	55	90	80	110
31	nidoqueen	90	92	87	75	85	76
34	nidoking	81	102	77	85	75	85
36	clefable	95	70	73	95	90	60
38	ninetales	73	76	75	81	100	100
40	wigglytuff	140	70	45	85	50	45

13. Kueri untuk menampilkan nama, total poin status, dan attack pokemon yang mempunyai attack di bawah rata-rata adalah

$$avg\_attack \leftarrow_{AVG(attack)} (\text{pokemon})$$

$$\Pi_{name,total\_base\_stats,attack}(\sigma_{attack<avg\_attack}(\text{pokemon}))$$

Kode SQL:

```
SELECT pokemon_name, total_base_stats, attack
FROM pokemon
WHERE attack > (SELECT AVG(attack) FROM pokemon);
```

Hasil yang diberikan: Terdapat total 538 Pokemon yang memiliki attack di bawah rata-rata

pokemon_name	total_base_stats	attack
bulbasaur	318	49
ivysaur	405	62
charmander	309	52
charmeleon	405	64
squirtle	314	48
wartortle	405	63
caterpie	195	30
metapod	205	20
butterfree	395	45
weedle	195	35
kakuna	205	25
pidgey	251	45

14. Kueri untuk menampilkan nama, total poin status, dan defense pokemon yang mempunyai defense di antara 100 dan 150 adalah

$$\Pi_{name,total\_base\_stats,defense}(\sigma_{100\leq defense\leq 150}(\text{pokemon}))$$

Kode SQL:

```
SELECT pokemon_name, total_base_stats, defense
FROM pokemon
WHERE defense BETWEEN 100 AND 150;
```

Hasil yang diberikan: Terdapat 170 pokemon yang memiliki defense antara 100 dan 150

pokemon_name	total_base_stats	defense
blastoise	530	100
sandslash	450	110
geodude	300	100
graveler	390	115
golem	495	130
slowbro	490	110
shellder	305	100
kingler	475	115
marowak	425	110
weezing	490	120

15. Kueri untuk menampilkan nama pokemon yang mempunyai total poin seluruh status di atas rata-rata dan attack di bawah rata-rata adalah

$$avg\_total \leftarrow_{AVG(total\_base\_stats)} (\text{pokemon})$$

$$avg\_attack \leftarrow_{AVG(attack)} (\text{pokemon})$$

$$\Pi_{name}(\sigma_{total\_base\_stats > avg\_total \wedge attack < avg\_attack}(\text{pokemon}))$$

Kode SQL:

```
SELECT pokemon_name
FROM pokemon
WHERE total_base_stats > (SELECT AVG(total_base_stats) FROM pokemon)
  AND attack < (SELECT AVG(attack) FROM pokemon);
```

Hasil yang diberikan: Terdapat 153 pokemon yang memiliki total poin seluruh status di atas rata-rata dan attack di bawah rata-rata

pokemon_name
clefable
ninetales
wigglytuff
venomoth
persian
alakazam
tentacruel
slowbro
magneton
dewgong

16. Misalkan setiap pokemon dapat dibagi menjadi 4 klasifikasi berdasarkan total base stats-nya, yaitu
    - Lemah: pokemon yang memiliki total base stats kurang dari 299
    - Sedang: pokemon yang memiliki total base stats antara 300-499
    - Kuat: pokemon yang memiliki total base stats antara 500-600
    - Sangat kuat: pokemon yang memiliki total base stats lebih dari 601.

    Total base stats digunakan sebagai metrik untuk mengukur seberapa kuat pokemon karena variabel ini merepresentasikan keseluruhan kekuatan dasar pokemon. Dengan demikian, jika mengikuti aturan yang sudah ada maka diperoleh hasilnya sebagai berikut

pokemon_name	total_base_stats	kategori
arceus	720	Sangat Kuat (OP/Legenda)
eternatus	690	Sangat Kuat (OP/Legenda)
mewtwo	680	Sangat Kuat (OP/Legenda)
lugia	680	Sangat Kuat (OP/Legenda)
ho-oh	680	Sangat Kuat (OP/Legenda)
rayquaza	680	Sangat Kuat (OP/Legenda)
dialga	680	Sangat Kuat (OP/Legenda)
palkia	680	Sangat Kuat (OP/Legenda)
giratina	680	Sangat Kuat (OP/Legenda)
reshiram	680	Sangat Kuat (OP/Legenda)
zekrom	680	Sangat Kuat (OP/Legenda)
xerneas	680	Sangat Kuat (OP/Legenda)
yveltal	680	Sangat Kuat (OP/Legenda)
solgaleo	680	Sangat Kuat (OP/Legenda)
lunala	680	Sangat Kuat (OP/Legenda)
slaking	670	Sangat Kuat (OP/Legenda)
kyogre	670	Sangat Kuat (OP/Legenda)

pokemon_name	total_base_stats	kategori
dragonite	600	Kuat
mew	600	Kuat
tyranitar	600	Kuat
celebi	600	Kuat
salamence	600	Kuat
metagross	600	Kuat
latias	600	Kuat
latios	600	Kuat
jirachi	600	Kuat
deoxys	600	Kuat
garchomp	600	Kuat
heatran	600	Kuat
manaphy	600	Kuat
darkrai	600	Kuat
shaymin	600	Kuat
victini	600	Kuat

pokemon_name	total_base_stats	kategori
talonflame	499	Sedang
drifblim	498	Sedang
simisage	498	Sedang
simisear	498	Sedang
simipour	498	Sedang
zebstrika	497	Sedang
golem	495	Sedang
magmar	495	Sedang
omastar	495	Sedang
kabutops	495	Sedang
cradily	495	Sedang
armaldo	495	Sedang
rampardos	495	Sedang
bastiodon	495	Sedang
floatzel	495	Sedang
mismagius	495	Sedang

pokemon_name	total_base_stats	kategori
vulpix	299	Lemah
tinkatink	297	Lemah
horsea	295	Lemah
shroomish	295	Lemah
electrike	295	Lemah
shuppet	295	Lemah
duskull	295	Lemah
blitzle	295	Lemah
tympole	294	Lemah
foongus	294	Lemah
munna	292	Lemah
sandile	292	Lemah
meowth	290	Lemah
pineco	290	Lemah
trapinch	290	Lemah
spheal	290	Lemah

Kode SQL yang digunakan:

```
SELECT pokemon_name, total_base_stats,
    CASE
      WHEN total_base_stats >= 601 THEN 'Sangat Kuat (OP/Legenda)'
      WHEN total_base_stats >= 500 THEN 'Kuat'
      WHEN total_base_stats >= 300 THEN 'Sedang'
      ELSE 'Lemah'
    END AS kategori
FROM pokemon
ORDER BY total_base_stats DESC;
```

17. Tidak selalu pokemon legendary di generasi sebelumnya pasti lebih lemah dibandingkan dengan generasi selanjutnya. Ada dua alasan yang mendukung pernyataan ini, antara lain:
    - Pertama, dengan membandingan total base stats milik pokemon berjenis legendary di setiap generasinya. Ambil contoh pada generasi ke-6 dan ke-7 seperti yang ditampilkan di cuplikan di bawah ini

generation	pokemon_name	total_base_stats
gen 6	xerneas	680
gen 6	yveltal	680
gen 6	zygarde	600
gen 7	solgaleo	680
gen 7	lunala	680
gen 7	necrozma	600
gen 7	silvally	570
gen 7	tapu-koko	570
gen 7	tapu-lele	570
gen 7	tapu-bulu	570
gen 7	tapu-fini	570
gen 7	cosmoem	400

      Dari gambar tersebut dapat dilihat bahwa untuk beberapa pokemon legendary yang muncul di generasi ke-7 memiliki total base stat yang lebih rendah dibandingkan dengan pokemon legendary yang muncul di generasi ke-6, contohnya total base stat yang dimiliki oleh pokemon zygarde, necrozma, silvally, tapu-koko, tapu-lele, tabu-bunu, tapu-fini, dan cosmoem di generasi ke-7 lebih rendah dibandingkan dengan total base stat pokemon milik xerneas dan yveltal di generasi ke-6.

      Kode SQL yang digunakan:

      ```
      SELECT generation, pokemon_name, total_base_stats
      FROM pokemon
      WHERE category = 'legendary'
      ORDER BY generation ASC, total_base_stats DESC;
      ```
    - Kedua, dilihat dari rata-rata total base stat dari setiap generasi. Perhatikan cuplikan berikut

generation	avg_tbs
gen 1	605.0000
gen 2	620.0000
gen 3	620.0000
gen 4	625.5556
gen 5	613.3333
gen 6	653.3333
gen 7	541.0000
gen 8	576.8182
gen 9	571.3636

Perhatikan bahwa dari generasi 1 sampai 4 memang terjadi kenaikan rata-rata total base stat namun pada generasi 5 terjadi penurunan dan meskipun rata-ratanya naik kembali pada generasi 6, di generasi ke-7 terjadi penurunan lagi. Dengan demikian, rata-rata total base stat dari generasi ke generasi belum tentu akan selalu naik. Oleh karena itu, alasan kedua ini merupakan salah satu alasan yang mendukung pernyataan awal.

Kode SQL yang digunakan:

```
SELECT generation, AVG(total_base_stats) AS avg_tbs
FROM pokemon
WHERE category = 'legendary'
GROUP BY generation
ORDER BY generation;
```

18. Tidak selalu pokemon bertipe legendary memiliki status individual yang tertinggi. Pernyataan ini didukung dengan alasan yang dapat dilihat pada cuplikan di bawah ini

pokemon_name	category	stat_type	max_stat
blissey	regular	HP	255
kartana	ultra beast	Attack	181
shuckle	regular	Defense	230
xurkitree	ultra beast	Special Attack	173
shuckle	regular	Special Defense	230
regieleki	legendary	Speed	200

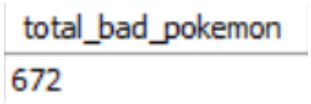
Perhatikan bahwa tabel tersebut menunjukkan nama-nama pokemon yang memiliki status individual tertinggi untuk masing-masing tipe status dan dapat dilihat bahwa pokemon bertipe legendary hanya terdapat satu, yaitu regieleki yang memiliki status speed tertinggi, namun untuk status seperti HP, defense, dan special defense justru pokemon dengan tipe regular yang menempati posisi tertinggi di ketiga status tersebut, sedangkan untuk status seperti attack dan special attack, pokemon bertipe ultra beast justru lebih unggul dibandingkan dengan legendary. Oleh karena itu, tidak selalu pokemon bertipe legendary yang memiliki status individual tertinggi.

Kode SQL yang digunakan:

```
SELECT pokemon_name, category, 'HP' AS stat_type, max_stat
FROM (SELECT pokemon_name, category, hp AS max_stat FROM pokemon ORDER BY hp DESC LIMIT 1) AS hp_max
UNION ALL
SELECT pokemon_name, category, 'Attack', max_stat
```

```
FROM (SELECT pokemon_name, category, attack AS max_stat FROM pokemon
ORDER BY attack DESC LIMIT 1) AS attack_max
UNION ALL
SELECT pokemon_name, category, 'Defense', max_stat
FROM (SELECT pokemon_name, category, defense AS max_stat FROM pokemon
ORDER BY defense DESC LIMIT 1) AS defense_max
UNION ALL
SELECT pokemon_name, category, 'Special Attack', max_stat
FROM (SELECT pokemon_name, category, special_attack AS max_stat FROM
pokemon ORDER BY special_attack DESC LIMIT 1) AS sp_atk_max
UNION ALL
SELECT pokemon_name, category, 'Special Defense', max_stat
FROM (SELECT pokemon_name, category, special_defense AS max_stat FROM
pokemon ORDER BY special_defense DESC LIMIT 1) AS sp_def_max
UNION ALL
SELECT pokemon_name, category, 'Speed', max_stat
FROM (SELECT pokemon_name, category, speed AS max_stat FROM pokemon
ORDER BY speed DESC LIMIT 1) AS speed_max;
```

19. Terdapat total 672 pokemon yang lebih payah daripada pokemon dengan attack tertinggi.



total_bad_pokemon
672

Kode SQL:

```
WITH strongest AS (
    SELECT pokemon_name, attack, special_attack
    FROM pokemon
    ORDER BY attack DESC
    LIMIT 1
)
SELECT COUNT(*) AS total_bad_pokemon
FROM pokemon p, strongest s
WHERE p.hp <= (s.attack + s.special_attack - p.defense - p.special_defense);
```

20. Persentase banyaknya pokemon yang lebih lincah daripada pokemon dengan defense tertinggi adalah 99,31%.

Kode SQL:

```
WITH most_defensive AS (
    SELECT defense, attack, speed
    FROM pokemon
    ORDER BY defense DESC
    LIMIT 1
),
```

```
faster_pokemon AS (
    SELECT COUNT(*) AS fast_count
    FROM pokemon p, most_defensive d
    WHERE p.attack > d.attack AND p.speed > d.speed
),
total_pokemon AS (
    SELECT COUNT(*) AS total_count FROM pokemon
)
SELECT
    (f.fast_count * 100.0 / t.total_count) AS percentage_faster
FROM faster_pokemon f, total_pokemon t;
```