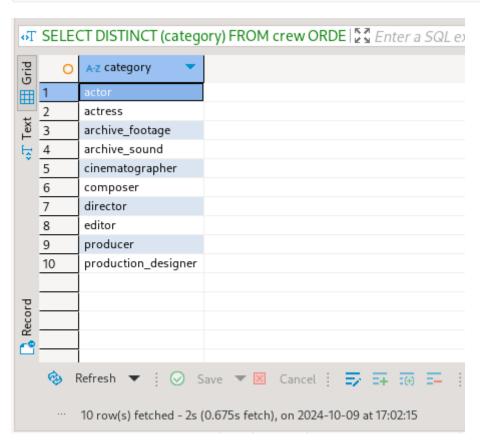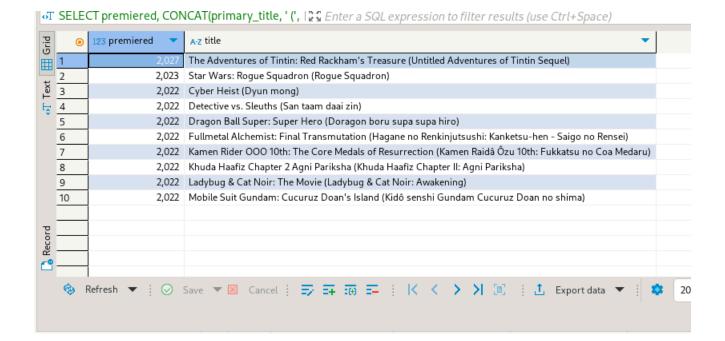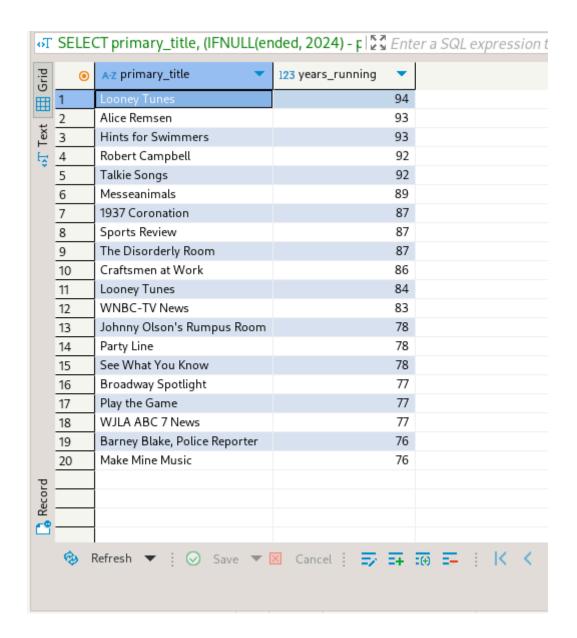# Database Query Engines

Q1

```sql
SELECT DISTINCT (category)
FROM crew
ORDER BY category
LIMIT 10;
```


Q2

```sql
SELECT premiered, CONCAT(primary_title, ' (', original_title, ')')
WHERE
    premiered IS NOT NULL
    AND genres LIKE "%Action%"
    AND type = "movie"
    AND original_title != primary_title
ORDER BY premiered DESC, primary_title
LIMIT 10;
```

123 premiered ▼	A-Z title ▼
1	2,027	The Adventures of Tintin: Red Rackham's Treasure (Untitled Adventures of Tintin Sequel)
2	2,023	Star Wars: Rogue Squadron (Rogue Squadron)
3	2,022	Cyber Heist (Dyun mong)
4	2,022	Detective vs. Sleuths (San taam daai zin)
5	2,022	Dragon Ball Super: Super Hero (Doragon boru supa supa hiro)
6	2,022	Fullmetal Alchemist: Final Transmutation (Hagane no Renkinjutsushi: Kanketsu-hen - Saigo no Rensei)
7	2,022	Kamen Rider OOO 10th: The Core Medals of Resurrection (Kamen Raidâ Ôzu 10th: Fukkatsu no Coa Medaru)
8	2,022	Khuda Haafiz Chapter 2 Agni Pariksha (Khuda Haafiz Chapter II: Agni Pariksha)
9	2,022	Ladybug & Cat Noir: The Movie (Ladybug & Cat Noir: Awakening)
10	2,022	Mobile Suit Gundam: Cucuruz Doan's Island (Kidô senshi Gundam Cucuruz Doan no shima)

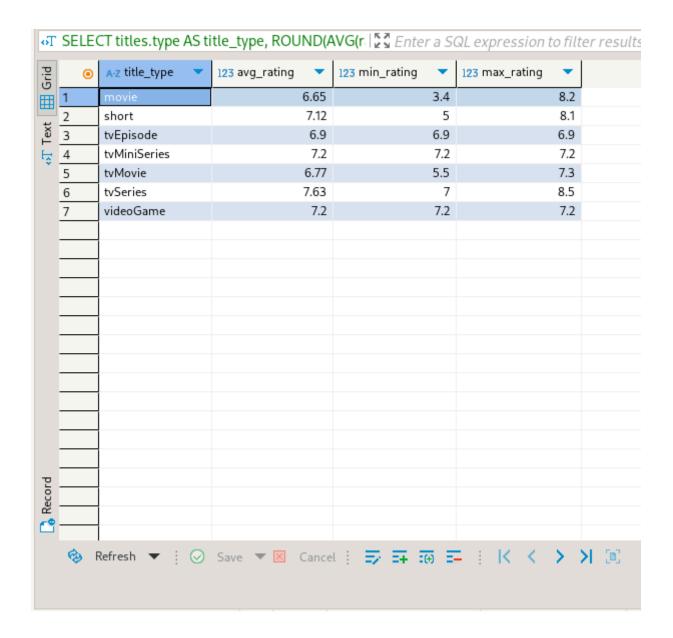Refresh ▼ ⊘ Save ▼ ⊠ Cancel ⇉ ⇥ ⊞ ☰ |< < > >| 🔲 ↥ Export data ▼ ⚙ 20

Q3

```
SELECT primary_title, (IFNULL(ended, 2024) - premiered) AS years_
WHERE
    premiered IS NOT NULL
    AND type = "tvSeries"
ORDER BY years_running DESC, primary_title
LIMIT 20;
```
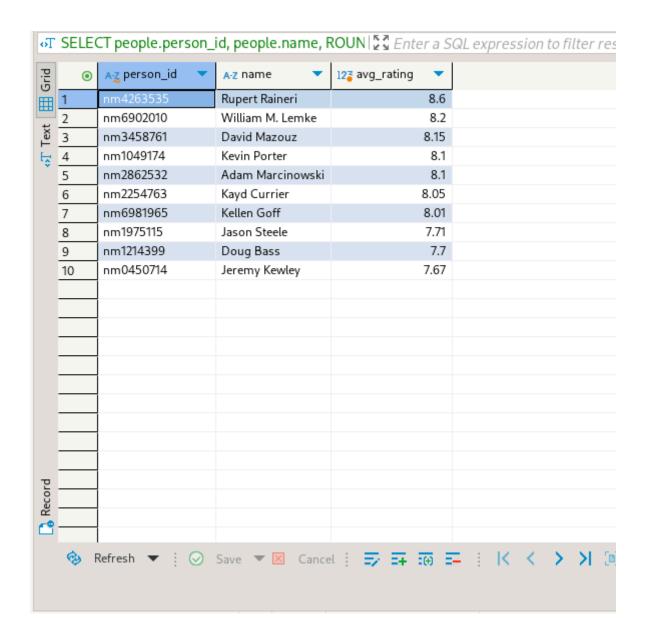
A-Z primary_title ▼	123 years_running ▼
1	Looney Tunes	94
2	Alice Remsen	93
3	Hints for Swimmers	93
4	Robert Campbell	92
5	Talkie Songs	92
6	Messeanimals	89
7	1937 Coronation	87
8	Sports Review	87
9	The Disorderly Room	87
10	Craftsmen at Work	86
11	Looney Tunes	84
12	WNBC-TV News	83
13	Johnny Olson's Rumpus Room	78
14	Party Line	78
15	See What You Know	78
16	Broadway Spotlight	77
17	Play the Game	77
18	WJLA ABC 7 News	77
19	Barney Blake, Police Reporter	76
20	Make Mine Music	76

Refresh ▼ ⋮ ⊘ Save ▼ ☒ Cancel ⋮ ⌰ ⌰ ⌰ ⌰ ⋮ |⟨ ‹

Q4

```
SELECT
    CONCAT(CAST(10 * ROUND(born / 10) as INTEGER), 's') AS decade,
    COUNT(DISTINCT people.person_id) as num_directors
FROM crew
JOIN people ON crew.person_id = people.person_id
WHERE
    category = "director"
    AND born IS NOT NULL
    AND (10 * ROUND(born / 10)) >= 1900
GROUP BY decade
ORDER BY decade;
```

«T SELECT CONCAT(CAST(10 * ROUND(born / 10) a| Enter a SQL expression

	decade	num_directors
1	1900s	376
2	1910s	389
3	1920s	721
4	1930s	810
5	1940s	999
6	1950s	1,084
7	1960s	1,625
8	1970s	2,151
9	1980s	1,884
10	1990s	745
11	2000s	86
12	2010s	1
13	2020s	1

Refresh ▼ ⊘ Save ▼ ⊠ Cancel ⠇ ⊟ ⊞ ⊡ ⊟ ⠇ K ‹

⋯ 13 row(s) fetched - 1s (0.071s fetch), on 2024-10-09 at 17:03:29

Q5

```
SELECT
    titles.type AS title_type,
    ROUND(AVG(rating),2) AS avg_rating,
    MIN(rating) AS min_rating,
    MAX(rating) AS max_rating
FROM akas
JOIN titles ON akas.title_id = titles.title_id
JOIN ratings ON akas.title_id = ratings.title_id
WHERE "language" = "de" AND (types = "imdbDisplay" OR types = "or:
GROUP BY titles.type;
```

SELECT titles.type AS title_type, ROUND(AVG(r... ⤢ *Enter a SQL expression to filter results*

title_type	avg_rating	min_rating	max_rating
1 movie	6.65	3.4	8.2
2 short	7.12	5	8.1
3 tvEpisode	6.9	6.9	6.9
4 tvMiniSeries	7.2	7.2	7.2
5 tvMovie	6.77	5.5	7.3
6 tvSeries	7.63	7	8.5
7 videoGame	7.2	7.2	7.2

Refresh ▼ ⋮ ⊘ Save ▼ ⊠ Cancel ⋮ ⫗ ⊒ ⫙ ⊟ ⋮ |< < > >| ▣

Q6

```
SELECT people.person_id, people.name, ROUND(AVG(rating), 2) AS avg
JOIN people ON people.person_id = crew.person_id
JOIN ratings ON crew.title_id = ratings.title_id
WHERE people.person_id IN (
    SELECT people.person_id FROM crew
    JOIN people ON people.person_id = crew.person_id
    WHERE crew.category = "actor" AND crew.characters LIKE "%Batma
)
GROUP BY people.person_id
ORDER BY avg_rating DESC
LIMIT 10;
```

SELECT people.person_id, people.name, ROUN⟨⟩ *Enter a SQL expression to filter res...*

#	A-Z person_id	A-Z name	123 avg_rating
1	nm4263535	Rupert Raineri	8.6
2	nm6902010	William M. Lemke	8.2
3	nm3458761	David Mazouz	8.15
4	nm1049174	Kevin Porter	8.1
5	nm2862532	Adam Marcinowski	8.1
6	nm2254763	Kayd Currier	8.05
7	nm6981965	Kellen Goff	8.01
8	nm1975115	Jason Steele	7.71
9	nm1214399	Doug Bass	7.7
10	nm0450714	Jeremy Kewley	7.67

Refresh ▼ ⋮ ⊘ Save ▼ ☒ Cancel ⋮ ⤳ ☴ ☵ ☶ ⋮ |< < > >| ▣

Q7

```
SELECT COUNT(DISTINCT person_id) from people WHERE born = (
    SELECT premiered FROM titles
    WHERE primary_title = "The Prestige" AND type = "movie"
);
```

Q8

Error in the question: `Rose%` includes names such as Rosemary, the actual query should be for `Rose %`

However, to match the sanity check in the question, `Rose%` is needed.

```sql
SELECT DISTINCT name FROM crew
JOIN people ON people.person_id = crew.person_id
WHERE title_id IN (
    SELECT DISTINCT title_id FROM crew
    JOIN people ON people.person_id = crew.person_id
    WHERE name LIKE "Rose%"
)
AND category = "director"
ORDER BY name;
```

⟨T SELECT DISTINCT name FROM crew JOIN peop| ⟨⟩ *Ente*

		A-Z name ▼
	1	Aimé Forget
	2	Alan Cooke
	3	Albert Zugsmith
	4	Alex Schead
	5	Alexander Gardiner
	6	Amber Edwards
	7	Amma Asante
	8	Angelica Cristina Dio
	9	Anthony Barajas
	10	Anthony Garner
	11	Anthony Parker
	12	Asad Panjwani
	13	Bill Hitchcock
	14	Bruce Kessler
	15	Bruno Piney
	16	Butch Hartman
	17	Bénédicte Delmas
	18	Cedric Messina
	19	Charles Beeson
	20	Christophe Douchand
	21	Clark Jones
	22	Colin Barr
	23	Cullen McGraw
	24	Daniel Leahey

Refresh ▼ ⋮ ⊘ Save ▼ ☒ Cancel ⋮ ⯊ ⯊ ⯊

Q9

```
SELECT category, name, died, primary_title, runtime_minutes, dead_
    SELECT
        ROW_NUMBER() OVER (
            PARTITION BY category
            ORDER BY died
        ) as dead_num,
        *
    FROM (
        SELECT
            ROW_NUMBER() OVER (
                PARTITION BY person_id
                ORDER BY runtime_minutes DESC, titles.title_id
            ) AS work_num,
            *
```

```
        FROM crew
        JOIN titles ON titles.title_id = crew.title_id
        WHERE runtime_minutes IS NOT NULL
    ) AS long_works
    JOIN people ON people.person_id = long_works.person_id
    WHERE work_num = 1 AND died IS NOT NULL
)
WHERE dead_num <= 5
ORDER BY category, died, name;
```
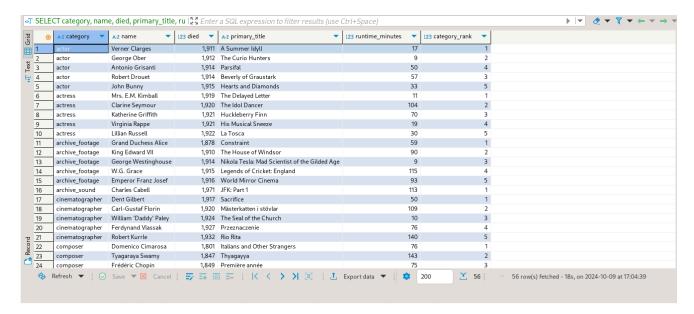
SELECT category, name, died, primary_title, ru Enter a SQL expression to filter results (use Ctrl+Space)

	category	name	died	primary_title	runtime_minutes	category_rank
1	actor	Verner Clarges	1,911	A Summer Idyll	17	1
2	actor	George Ober	1,912	The Curio Hunters	9	2
3	actor	Antonio Grisanti	1,914	Parsifal	50	4
4	actor	Robert Drouet	1,914	Beverly of Graustark	57	3
5	actor	John Bunny	1,915	Hearts and Diamonds	33	5
6	actress	Mrs. E.M. Kimball	1,919	The Delayed Letter	11	1
7	actress	Clarine Seymour	1,920	The Idol Dancer	104	2
8	actress	Katherine Griffith	1,921	Huckleberry Finn	70	3
9	actress	Virginia Rappe	1,921	His Musical Sneeze	19	4
10	actress	Lillian Russell	1,922	La Tosca	30	5
11	archive_footage	Grand Duchess Alice	1,878	Constraint	59	1
12	archive_footage	King Edward VII	1,910	The House of Windsor	90	2
13	archive_footage	George Westinghouse	1,914	Nikola Tesla: Mad Scientist of the Gilded Age	9	3
14	archive_footage	W.G. Grace	1,915	Legends of Cricket: England	115	4
15	archive_footage	Emperor Franz Josef	1,916	World Mirror Cinema	93	5
16	archive_sound	Charles Cabell	1,971	JFK: Part 1	113	1
17	cinematographer	Dent Gilbert	1,917	Sacrifice	50	1
18	cinematographer	Carl-Gustaf Florin	1,920	Mästerkatten i stövlar	109	2
19	cinematographer	William 'Daddy' Paley	1,924	The Seal of the Church	10	3
20	cinematographer	Ferdynand Vlassak	1,927	Przeznaczenie	76	4
21	cinematographer	Robert Kurrle	1,932	Rio Rita	140	5
22	composer	Domenico Cimarosa	1,801	Italians and Other Strangers	76	1
23	composer	Tyagaraya Swamy	1,847	Thyagayya	143	2
24	composer	Frédéric Chopin	1,849	Première année	75	3

Refresh ▼ ⊘ Save ▼ ✕ Cancel |K < > >| | ↥ Export data ▼ ⚙ 200 ⤓ 56 ⋯ 56 row(s) fetched - 18s, on 2024-10-09 at 17:04:39

Q10

```
SELECT GROUP_CONCAT(atom) FROM (
    SELECT DISTINCT atom FROM crew, json_each(crew.characters)
    JOIN people ON crew.person_id = people.person_id
    WHERE name = "Leonardo DiCaprio" AND atom NOT LIKE "%Self%"
);
```

SELECT GROUP_CONCAT(atom) FROM (SELEC Enter a SQL expression to filter results (use Ctrl+Space)

	GROUP_CONCAT(atom)
1	Luke Brower,Garry Buckman,Toby,Arnie Grape,Bud,Jim Carroll,Kid,Arthur Rimbaud,Hank,Romeo,Derek,Jack Dawson,Brandon Darrow,King Louis,Phillippe,Richard,Amsterdam Vallon,Frank Abagnale Jr.,Howard Hughes,Billy,Danny