

```
1  -- Find average rating for each book
2  SELECT "book_id", ROUND(AVG("rating"), 2) AS "average rating" FROM "ratings"
3  GROUP BY "book_id";
4
5  -- Join titles
6  SELECT "title", ROUND(AVG("rating"), 2) AS "average rating" FROM "ratings"
7  JOIN "books" ON "books"."id" = "ratings"."book_id"
8  GROUP BY "book_id";
9
10 -- Choosing books with a rating of 4.0 or higher
11 SELECT "title", ROUND(AVG("rating"), 2) AS "average rating" FROM "ratings"
12 JOIN "books" ON "books"."id" = "ratings"."book_id"
13 GROUP BY "book_id"
14 HAVING "average rating" > 4.0;
```

```
1  -- Show all sea lions for which we have data
2  SELECT * FROM "sea_lions"
3  JOIN "migrations" ON "migrations"."id" = "sea_lions"."id";
4
5  -- Show all sea lions, whether or not we have data
6  SELECT * FROM "sea_lions"
7  LEFT JOIN "migrations" ON "migrations"."id" = "sea_lions"."id";
8
9  -- Show all data, whether or not there are matching sea lions
10 SELECT * FROM "sea_lions"
11 RIGHT JOIN "migrations" ON "migrations"."id" = "sea_lions"."id";
12
13 -- Show all data and all sea lions
14 SELECT * FROM "sea_lions"
15 FULL JOIN "migrations" ON "migrations"."id" = "sea_lions"."id";
16
17 -- JOIN sea lions and migrations without specifying matching column
18 SELECT * FROM "sea_lions"
19 NATURAL JOIN "migrations";
20
21 -- Use WHERE after joining a table
22 SELECT * FROM "sea_lions"
23 JOIN "migrations" ON "migrations"."id" = "sea_lions"."id"
24 WHERE "migrations"."distance" > 1500;
```

```
1  -- Find all books published by MacLehose Press, with hard-coded id
2  SELECT "id" FROM "publishers" WHERE "publisher" = 'MacLehose Press';
3
4  SELECT "title" FROM "books" WHERE "publisher_id" = 12;
5
6  -- Find all books published by MacLehose Press, with a nested query
7  SELECT "title" FROM "books" WHERE "publisher_id" = (
8      SELECT "id" FROM "publishers" WHERE "publisher" = 'MacLehose Press'
9  );
10
11 -- Find all ratings for "In Memory of Memory"
12 SELECT "rating" FROM "ratings" WHERE "book_id" = (
13     SELECT "id" FROM "books" WHERE "title" = 'In Memory of Memory'
14 );
15
16 -- Find average rating for "In Memory of Memory"
17 SELECT AVG("rating") FROM "ratings" WHERE "book_id" = (
18     SELECT "id" FROM "books" WHERE "title" = 'In Memory of Memory'
19 );
20
21 -- Which author wrote "The Birthday Party"?
22 SELECT "id" FROM "books" WHERE "title" = 'The Birthday Party';
23
24 SELECT "author_id" FROM "authored" WHERE "book_id" = (
25     SELECT "id" FROM "books" WHERE "title" = 'The Birthday Party'
26 );
27
28 SELECT "name" FROM "authors" WHERE "id" = (
29     SELECT "author_id" FROM "authored" WHERE "book_id" = (
30         SELECT "id" FROM "books" WHERE "title" = 'The Birthday Party'
31     )
32 );
33
34 -- Find all books by Fernanda Melchor, using IN
35 SELECT "id" FROM "authors" WHERE "name" = 'Fernanda Melchor';
36
37 SELECT "book_id" FROM "authored" WHERE "author_id" = (
38     SELECT "id" FROM "authors" WHERE "name" = 'Fernanda Melchor'
39 );
40
41 SELECT "title" FROM "books" WHERE "id" IN (
42     SELECT "book_id" FROM "authored" WHERE "author_id" = (
```

```
43         SELECT "id" FROM "authors" WHERE "name" = 'Fernanda Melchor'
44     )
45 );
46
47 -- Using IN to search for multiple authors
48 SELECT "title" FROM "books" WHERE "id" IN (
49     SELECT "book_id" FROM "authored" WHERE "author_id" IN (
50         SELECT "id" FROM "authors" WHERE "name" IN ('Fernanda Melchor', 'Annie Ernaux')
51     )
52 );
```

```
1  -- UNION
2
3  -- Select all authors, labeling as authors
4  SELECT 'author' AS "profession", "name" FROM "authors";
5
6  -- Select all translators, labeling as translators
7  SELECT 'translator' AS "profession", "name" FROM "translators";
8
9  -- Combine authors and translators into one result set
10 SELECT 'author' AS "profession", "name" FROM "authors";
11 UNION
12 SELECT 'translator' AS "profession", "name" FROM "translators";
13
14 -- INTERSECT
15 -- Assume names are unique
16
17 -- Find authors and translators
18 SELECT "name" FROM "authors"
19 INTERSECT
20 SELECT "name" FROM "translators";
21
22 -- Find books translated by Sophie Hughes
23 SELECT "book_id" FROM "translated" WHERE "translator_id" = (
24     SELECT "id" FROM "translators" WHERE name = 'Sophie Hughes'
25 );
26
27 -- Find books translated by Margaret Jull Costa
28 SELECT "book_id" FROM "translated" WHERE "translator_id" = (
29     SELECT "id" FROM "translators" WHERE name = 'Margaret Jull Costa'
30 );
31
32 -- Find intersection of books
33 SELECT "book_id" FROM "translated" WHERE "translator_id" = (
34     SELECT "id" FROM "translators" WHERE name = 'Sophie Hughes'
35 )
36 INTERSECT
37 SELECT "book_id" FROM "translated" WHERE "translator_id" = (
38     SELECT "id" FROM "translators" WHERE name = 'Margaret Jull Costa'
39 );
40
41 -- Find intersection of books
42 SELECT "title" FROM "books" WHERE "id" = (
```

```
43     SELECT "book_id" FROM "translated" WHERE "translator_id" = (  
44     SELECT "id" FROM "translators" WHERE name = 'Sophie Hughes'  
45     )  
46     INTERSECT  
47     SELECT "book_id" FROM "translated" WHERE "translator_id" = (  
48         SELECT "id" FROM "translators" WHERE name = 'Margaret Jull Costa'  
49     )  
50 );  
51  
52 -- EXCEPT  
53 -- Assume names are unique  
54  
55 -- Find translators who are not authors  
56 SELECT "name" FROM "translators"  
57 EXCEPT  
58 SELECT "name" FROM "authors";
```