SQL in two tables

Authors table

id	last_name	first_name	DoB	Income	Genre
1	Lopez Baranda	Christina	15/11/2000	55000	Fantasy
2	Jin-Soon	Sin	29/03/1983	65000	Crime
3	Jones	Hannah	01/02/1973	129000	Fantasy
4	Novak	Stanislaw	12/12/1992	91000	Crime
5	Turay	Tandice	09/07/1980	99000	Romance
6	Roy	Shanta	11/10/1977	55000	Fantasy
7	Berger	Henry	15/08/1956	63000	Romance
8	Khatami	Paree	11/10/1966	86000	Sci-Fi

Books table

id	title	ISBN
1	Creating relational databases for fun and profit	7654321123456
2	Relational databases for really smart people	9876543212345
3	My life with relational databases: a memoir	3212345678909
4	Relational databases: an existential journey	8172635412345

BooksAuthors table

book_id	author_id
3	6
2	4
2	5
1	1
1	3
1	5
4	8

Editions table

edition_id	book_id	date_of_publication	edition_number
1	3	2001	1
2	3	2003	2
3	4	2003	1
5	1	2000	1
6	3	2005	3
8	2	2012	1
9	3	2009	4

Foreign key

edition_id	book_id	date_of_publication	edition_number
1	3	2001	1
2	3	2003	2
3	4	2003	1
5	1	2000	1
6	3	2005	3
8	2	2012	1
9	3	2009	4

- Each edition is related to a book
- book_id is a **foreign key** that refers to books
- Each *non null* values of book_id must be found in the id column of books

Enforcing a foreign key

What if a book is deleted or its id changed?

We have 3 choices, decided when the constraint is created:

1. delete/change also the edition (CASCADE)
2. abort the operation
3. set the book_id to NULL (SET NULL)

All choices guarantee the integrity of the database

Find the foreign key

Books

id	title	ISBN
1	Creating relational databases for fun and profit	7654321123456
2	Relational databases for really smart people	9876543212345

BooksAuthors

book_id	author_id
2	4
2	5
1	1

Query

This query asks for the first and last names of authors of the book with id 1:

The results are:

first_name	last_name
Hannah	Jones
Christina	Lopez Baranda
Tandice	Turay

Query (where version)

```sql
SELECT first_name, last_name
FROM Authors, BooksAuthors
WHERE BooksAuthors.author_id = Authors.author_id
AND book_id = 1;
```

Query (join version)

```sql
SELECT first_name, last_name
FROM Authors JOIN BooksAuthors
    ON BooksAuthors.author_id = Authors.author_id
WHERE book_id = 1;
```

The JOIN version is better when only 2 tables are involved

Query

To find the book IDs and ISBNs that have editions published after (that is, greater than) 2003.

id	ISBN	date_of_publication
2	9876543212345	2012
3	3212345678909	2005
3	3212345678909	2009

```
SELECT Books.id, ISBN, date_of_publication
FROM Books, Editions
WHERE Books.id = Editions.book_id
AND Editions.date_of_publication > 2003;
```

Query results

id	title	ISBN
2	Relational databases for really smart people	9876543212345
3	My life with relational databases: a memoir	3212345678909
3	My life with relational databases: a memoir	3212345678909

```sql
SELECT id, title, ISBN
FROM Books, Editions
WHERE Books.id = Editions.book_id
AND Editions.date_of_publication > 2003;
```

1. Duplicate rows

Query results

id	title	ISBN
2	Relational databases for really smart people	9876543212345
3	My life with relational databases: a memoir	3212345678909

```
SELECT DISTINCT id, title, ISBN
FROM Books, Editions
WHERE Books.id = Editions.book_id
AND Editions.date_of_publication > 2003;
```

Query

Find who has written a book whose ISBN ends with 5

id	ISBN	id	last_name	first_name
2	9876543212345	4	Novak	Stanislaw
2	9876543212345	5	Turay	Tandice
4	8172635412345	8	Khatami	Paree

```
SELECT Books.id, ISBN, Authors.id,
       last_name, first_name
FROM Books, Authors, BooksAuthors
WHERE Books.id = BooksAuthors.book_id AND
      BooksAuthors.author_id = Authors.id AND
      isbn LIKE "%5";
```

Query

```sql
SELECT Books.id, ISBN, Authors.id,
       last_name, first_name
FROM Books, Authors, BooksAuthors
WHERE Books.id = BooksAuthors.book_id AND
      BooksAuthors.author_id = Authors.id AND
      isbn LIKE "%5";
```

1. There are three tables involved
2. id is a column name of two tables, use the table to disambiguate

Condition on more tables

Find who has a last name with exactly 5 characters and has written a book whose ISBN ends with 5

id	ISBN	id	last_name	first_name
2	9876543212345	4	Novak	Stanislaw
2	9876543212345	5	Turay	Tandice

```
SELECT Books.id, ISBN, Authors.id,
       last_name, first_name
FROM Books, Authors, BooksAuthors
WHERE Books.id = BooksAuthors.book_id AND
      BooksAuthors.author_id = Authors.id AND
      last_name LIKE '_____' AND
      isbn LIKE ”%5”;
```

Table aliases

```
SELECT Books.id, ISBN, Authors.id,
       last_name, first_name
FROM Books, Authors, BooksAuthors
WHERE Books.id = BooksAuthors.book_id AND
      BooksAuthors.author_id = Authors.id AND
      last_name LIKE '_____' AND
      isbn LIKE "%5";

SELECT b.id, ISBN, a.id,
       last_name, first_name
FROM Books b, Authors a , BooksAuthors ba
WHERE b.id = ba.book_id AND
      ba.author_id = a.id AND
      last_name LIKE '_____' AND
      isbn LIKE "%5";
```

Join Semantics

```
SELECT T1.A
FROM T1, T2
WHERE T1.B=T2.C;
```

1. Cross product between T1 and T2
2. Select only the rows satisfying the WHERE clause
3. Projection on the A column

NULL affects queries

For each author, list the ISBN of the books they have written

```
SELECT Books.id, ISBN, Authors.id,
       last_name, first_name
FROM Books, Authors, BooksAuthors
WHERE Books.id = BooksAuthors.book_id AND
      BooksAuthors.author_id = Authors.id;
```

But the author with id 2 has written no books

NULL affects queries

id	ISBN	id	last_name	first_name
3	3212345678909	6	Roy	Shanta
2	9876543212345	4	Novak	Stanislaw
2	9876543212345	5	Turay	Tandice
1	7654321123456	1	Lopez Baranda	Christina
1	7654321123456	3	Jones	Hannah
1	7654321123456	5	Turay	Tandice
4	8172635412345	8	Khatami	Paree

Outer Join

```
SELECT Authors.id, last_name, first_name
       Books.id, ISBN,
FROM Books, Authors LEFT JOIN BooksAuthors
    ON Authors.id=BooksAuthors.author_id
WHERE Books.id = BooksAuthors.book_id;
```

Outer Join

id	last_name	first_name	id	ISBN
6	Roy	Shanta	3	3212345678909
4	Novak	Stanislaw	2	9876543212345
5	Turay	Tandice	2	9876543212345
1	Lopez Baranda	Christina	1	7654321123456
3	Jones	Hannah	1	7654321123456
5	Turay	Tandice	1	7654321123456
8	Khatami	Paree	4	8172635412345
2	Jin-Soon	Sin	NULL	NULL
7	Berger	Henry	NULL	NULL

Counting

For each author, find the number of books they have written

```
SELECT Authors.id, last_name, first_name,
       count(*) as number
FROM Books, Authors, BooksAuthors
WHERE Books.id = BooksAuthors.book_id AND
      BooksAuthors.author_id = Authors.id
GROUP BY Authors.id;
```

Counting

For each author, find the number of books they have written

```
SELECT Authors.id, last_name, first_name,
       count(*) as number
FROM Books, Authors, BooksAuthors
WHERE Books.id = BooksAuthors.book_id AND
      BooksAuthors.author_id = Authors.id
GROUP BY Authors.id;
```

NO

The variables in the SELECT and in the GROUP BY clauses must be consistent

Counting

For each author, find the number of books they have written

```
SELECT Authors.id, last_name, first_name,
       count(*) as number
FROM Books, Authors, BooksAuthors
WHERE Books.id = BooksAuthors.book_id AND
      BooksAuthors.author_id = Authors.id
GROUP BY Authors.id, last_name, first_name;
```

Result

id	last_name	first_name	number
6	Roy	Shanta	1
4	Novak	Stanislaw	1
5	Turay	Tandice	2
1	Lopez Baranda	Christina	1
3	Jones	Hannah	1
8	Khatami	Paree	1

Counting

For each author, find the number of books they have written

```
SELECT Authors.id, last_name, first_name,
       count(*) as number
FROM Books, Authors LEFT JOIN BooksAuthors
    ON Authors.id=BooksAuthors.author_id
WHERE Books.id = BooksAuthors.book_id
GROUP BY Authors.id, last_name, first_name,;
```

count(Books.id) counts the number of tuples where Books.id is not NULL

Result

id	last_name	first_name	number
6	Roy	Shanta	1
4	Novak	Stanislaw	1
5	Turay	Tandice	2
1	Lopez Baranda	Christina	1
3	Jones	Hannah	1
8	Khatami	Paree	1
2	Jin-Soon	Sin	0
7	Berger	Henry	0

License