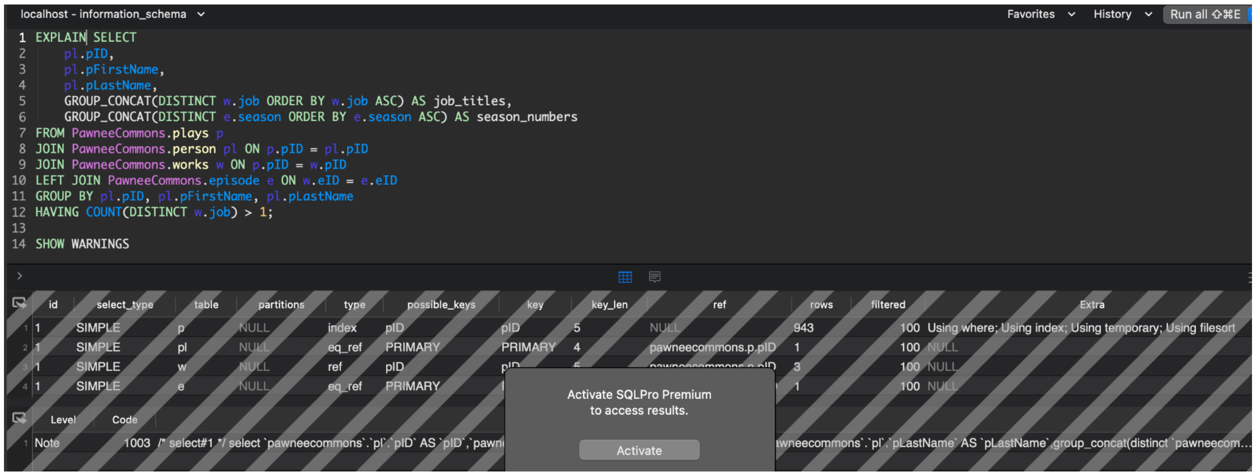
Generating Explain



localhost - information_schema

```sql
EXPLAIN SELECT
    pl.pID,
    pl.pFirstName,
    pl.pLastName,
    GROUP_CONCAT(DISTINCT w.job ORDER BY w.job ASC) AS job_titles,
    GROUP_CONCAT(DISTINCT e.season ORDER BY e.season ASC) AS season_numbers
FROM PawneeCommons.plays p
JOIN PawneeCommons.person pl ON p.pID = pl.pID
JOIN PawneeCommons.works w ON p.pID = w.pID
LEFT JOIN PawneeCommons.episode e ON w.eID = e.eID
GROUP BY pl.pID, pl.pFirstName, pl.pLastName
HAVING COUNT(DISTINCT w.job) > 1;

SHOW WARNINGS
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	p	NULL	index	pID	pID	5	NULL	943	100	Using where; Using index; Using temporary; Using filesort
1	SIMPLE	pl	NULL	eq_ref	PRIMARY	PRIMARY	4	pawneecommons.p.pID	1	100	NULL
1	SIMPLE	w	NULL	ref	pID	pID	5	pawneecommons.p.pID	3	100	NULL
1	SIMPLE	e	NULL	eq_ref	PRIMARY	[illegible]	[illegible]	[illegible]	1	100	NULL

Level	Code	Message
Note	1003	/* select#1 */ select `pawneecommons`.`pl`.`pID` AS `pID`,`pawn[illegible]wneecommons`.`pl`.`pLastName` AS `pLastName`,group_concat(distinct `pawneecom…

Activate SQLPro Premium to access results.

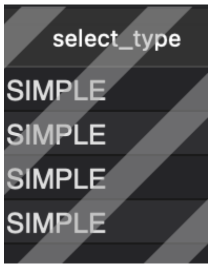
Activate

MySQL actually executes the query. But at each JOIN, instead of executing, it fills the EXPLAIN result set.

What is a JOIN?
Everything is a join , because MySQL always uses nested-loops. Even a single-table SELECT or a UNION or a subquery.

➢ The “**id**” column shows which SELECT the row belongs to : if only SELECT with no subquery or UNION, then 1 like our result.(Otherwise, generally numbered sequentially)

Simple/Complex :simple: there is only SELECT in the whole query but the



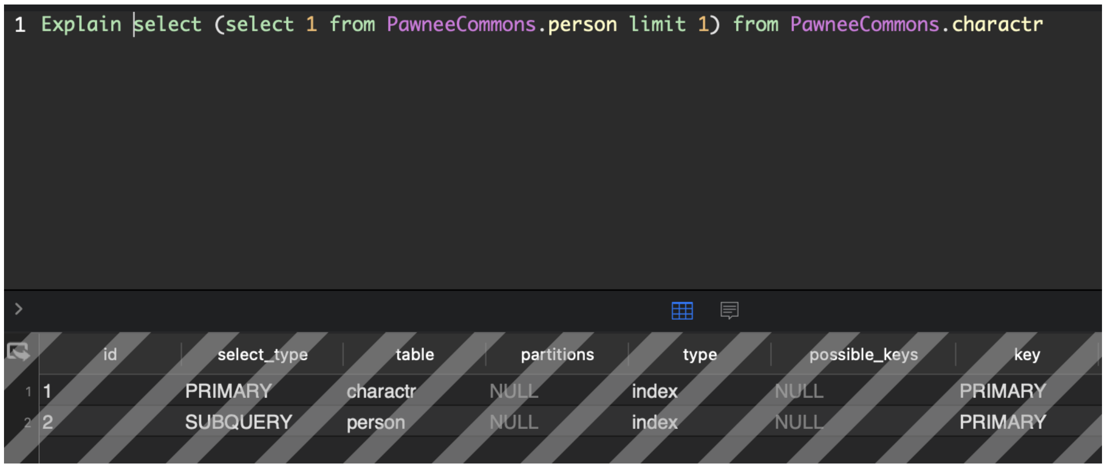
select_type
SIMPLE
SIMPLE
SIMPLE
SIMPLE

Complex:(3 subtypes)

subquery: numbered according to position in SQL text

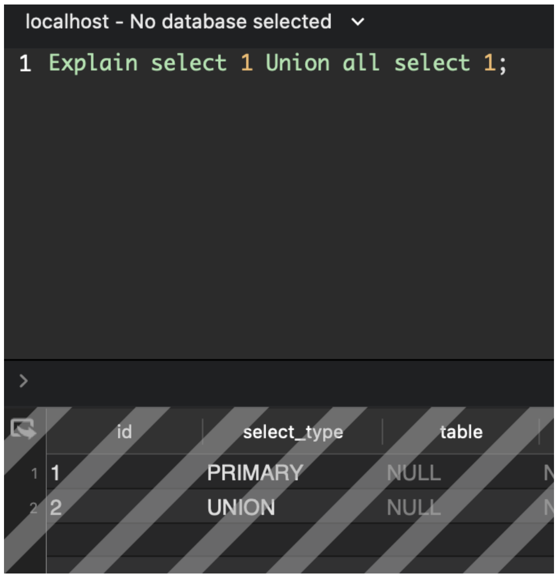
derived: executed as a temp table

union: fill a temp table, then read out with a NULL id in a row that says UNION RESULT



```
Explain select (select 1 from PawneeCommons.person limit 1) from PawneeCommons.charactr
```

	id	select_type	table	partitions	type	possible_keys	key
1	1	PRIMARY	charactr	NULL	index	NULL	PRIMARY
2	2	SUBQUERY	person	NULL	index	NULL	PRIMARY



localhost - No database selected

```
Explain select 1 Union all select 1;
```

	id	select_type	table
1	1	PRIMARY	NULL
2	2	UNION	NULL

- The “**select_type**” column shows Simple vs. Complex(which type of complex)
  Special UNION rule: first contained SELECT matches outer context
  Dependencies, un-cacheability : refers to the item_cache, not query cache

- The “**table**” column shows the table name or its alias, derived tables are complicated, forward reference :<derivedN>. UNION is complicated too ,backward references : <Union 1,2,3,…>

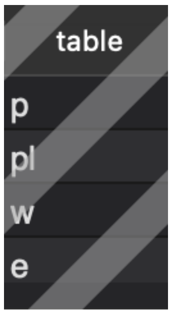
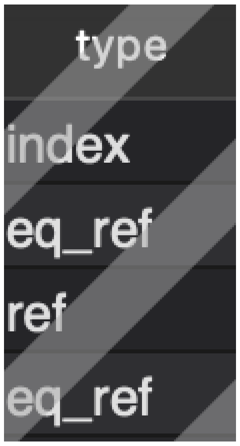


table
p
pl
w
e

- The “**Type**” column shows how MySQL will access rows.
  Worse to better:
  All, index, range, ref, eq_ref, const, system, NULL

FROM PawneeCommons.plays p	 type index eq_ref ref eq_ref
JOIN PawneeCommons.person pl ON p.pID = pl.pID	
JOIN PawneeCommons.works w ON p.pID = w.pID HAVING COUNT(DISTINCT w.job) > 1;	
LEFT JOIN PawneeCommons.episode e ON w.eID = e.eID	

- **All**:A full table scan is done for each combination of rows from the previous tables. This is normally not good if the table is the first table not marked const, and usually very bad in all other cases.
- **index**: The index join type is the same as ALL, except that the index tree is scanned. This occurs two ways:
  If the index is a covering index for the queries and can be used to satisfy all data required from the table, only the index tree is scanned.
  In this case, the Extra column says Using index. An index-only scan usually is faster than ALL because the size of the index usually is smaller than the table data.
  A full table scan is performed using reads from the index to look up data rows in index order. Uses index does not appear in the Extra column.

- **range** : range can be used when a key column is compared to a constant using any of the =, <>, >, >=, <, <=, IS NULL, <=>, BETWEEN, LIKE, or IN() operators

- **ref:** ref can be used for indexed columns that are compared using the = or <=> operator.

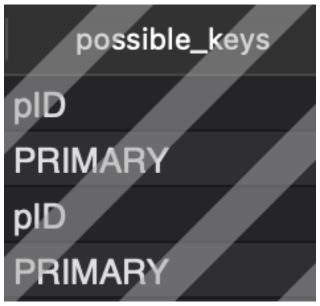
- **eq_ref**: eq_ref can be used for indexed columns that are compared using the = operator. The comparison value can be a constant or an expression that uses columns from tables that are read before this table.

- **constants** : The table has at most one matching row, which is read at the start of the query. Because there is only one row, values from the column in this row can be regarded as constants by the rest of the optimizer. const tables are very fast because they are read only once. const is used when you compare all parts of a PRIMARY KEY or UNIQUE index to constant values.

- **system**: The table has only one row (= system table). This is a special case of the const join type.

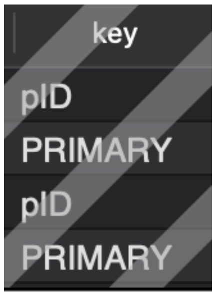
NULL: no table involved.

The “**Possible_keys**” columns show which indexes were candidates.



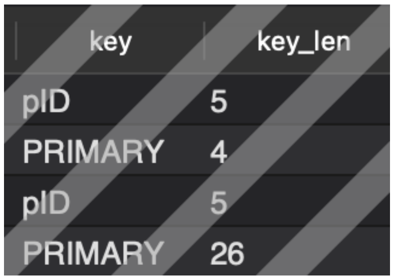
possible_keys
pID
PRIMARY
pID
PRIMARY

The “**key**” column shows which indexes the optimizer chose.



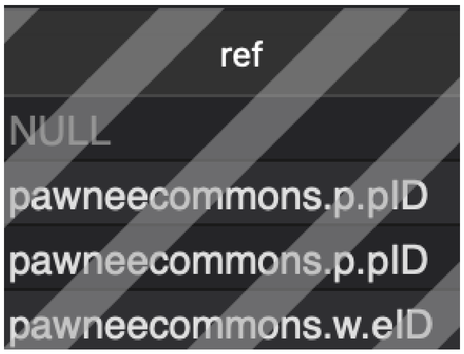
key
pID
PRIMARY
pID
PRIMARY

The “**key_len**” column shows that How many bytes of the index will be used.



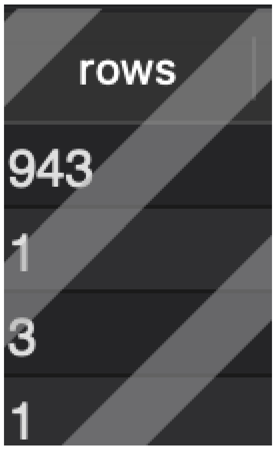
key	key_len
pID	5
PRIMARY	4
pID	5
PRIMARY	26

The “**Ref**” column shows the source of values used for lookups.The ref column shows which columns or constants are compared to the index named in the key column to select rows from the table.



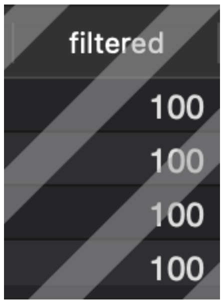
ref
NULL
pawneecommons.p.pID
pawneecommons.p.pID
pawneecommons.w.eID

The “**rows**” column estimated #rows to examine in the table/index.



rows
943
1
3
1

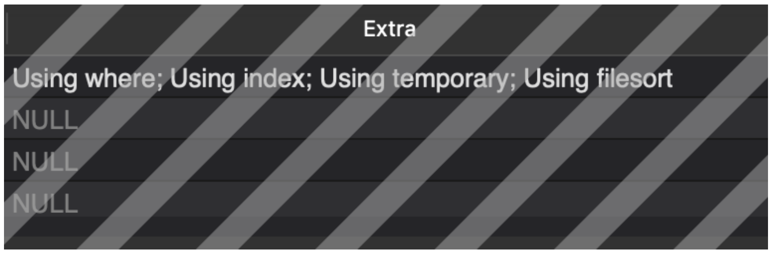
The”**filtered**” column shows the percentage of rows that satisfy WHERE, usually 0 or 100 ; complex behavior.



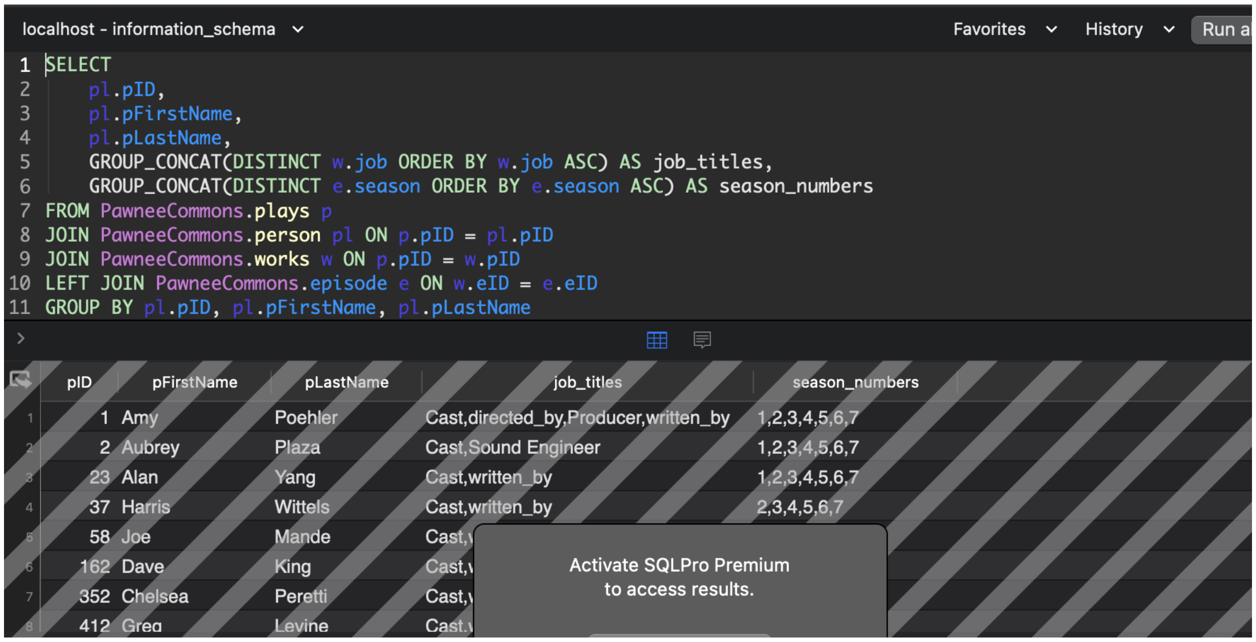
filtered
100
100
100
100

The “**Extra”** column of EXPLAIN output contains additional information about how MySQL resolves the query. The following list explains the values that can appear in this column. Each item also indicates for JSON-formatted output which property displays the

Extra value. For some of these, there is a specific property. The others display as the text of the message property.
If you want to make your queries as fast as possible, look out for Extra column values of Using filesort and Using temporary, or, in JSON-formatted EXPLAIN output, for using_filesort and using_temporary_table properties equal to true.



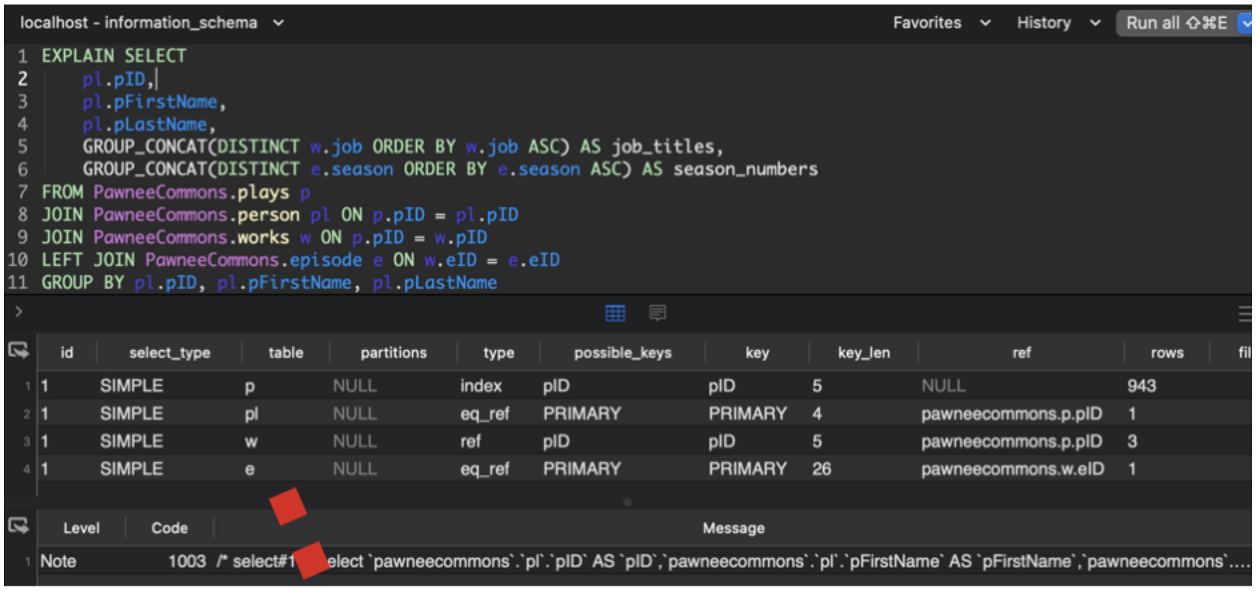
Extra
Using where; Using index; Using temporary; Using filesort
NULL
NULL
NULL



```sql
SELECT
    pl.pID,
    pl.pFirstName,
    pl.pLastName,
    GROUP_CONCAT(DISTINCT w.job ORDER BY w.job ASC) AS job_titles,
    GROUP_CONCAT(DISTINCT e.season ORDER BY e.season ASC) AS season_numbers
FROM PawneeCommons.plays p
JOIN PawneeCommons.person pl ON p.pID = pl.pID
JOIN PawneeCommons.works w ON p.pID = w.pID
LEFT JOIN PawneeCommons.episode e ON w.eID = e.eID
GROUP BY pl.pID, pl.pFirstName, pl.pLastName
```

	pID	pFirstName	pLastName	job_titles	season_numbers
1	1	Amy	Poehler	Cast,directed_by,Producer,written_by	1,2,3,4,5,6,7
2	2	Aubrey	Plaza	Cast,Sound Engineer	1,2,3,4,5,6,7
3	23	Alan	Yang	Cast,written_by	1,2,3,4,5,6,7
4	37	Harris	Wittels	Cast,written_by	2,3,4,5,6,7
5	58	Joe	Mande	Cast,	
6	162	Dave	King	Cast,	
7	352	Chelsea	Peretti	Cast,	
8	412	Greg	Levine	Cast,	

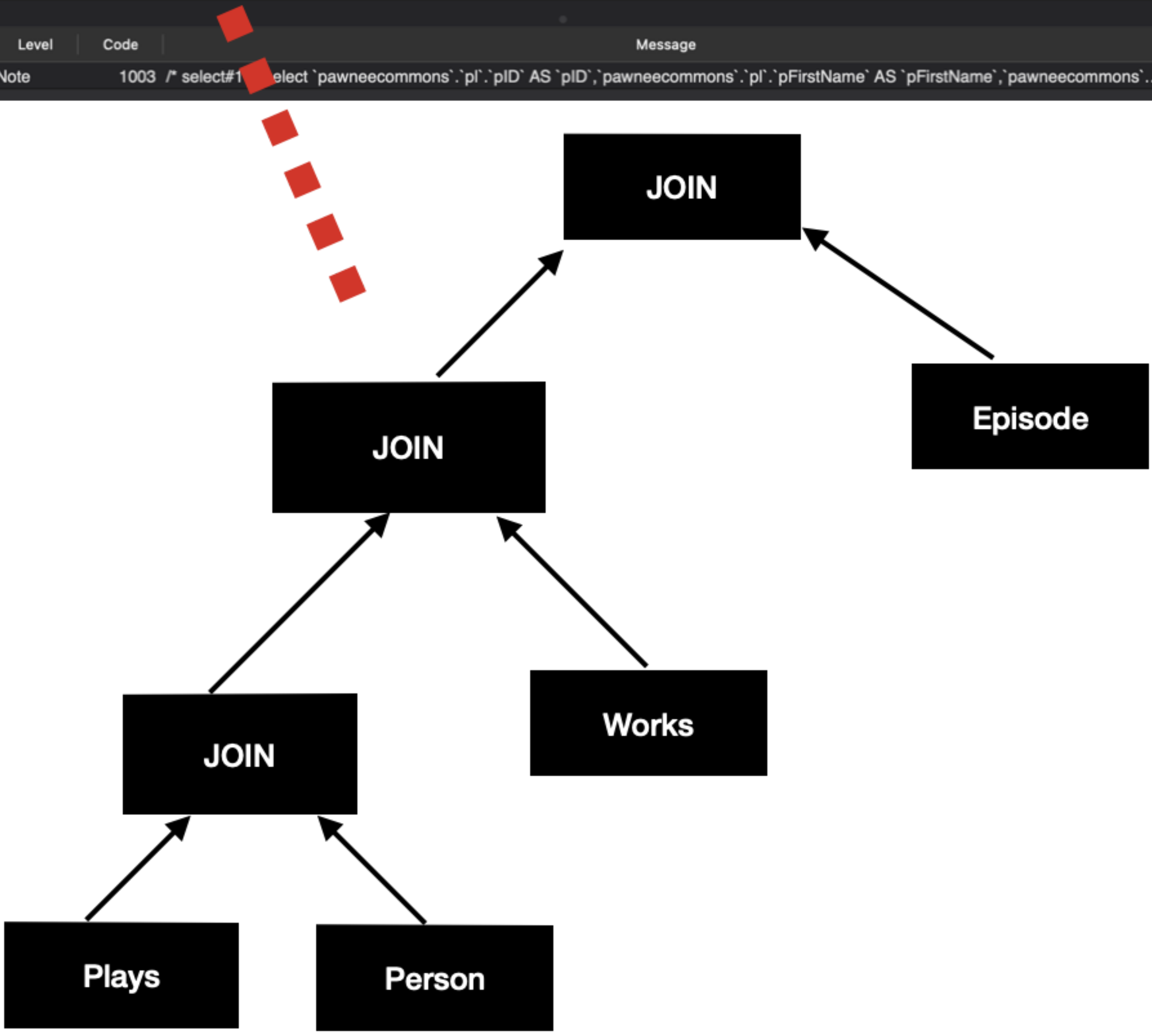
Activate SQLPro Premium
to access results.



```
EXPLAIN SELECT
    pl.pID,
    pl.pFirstName,
    pl.pLastName,
    GROUP_CONCAT(DISTINCT w.job ORDER BY w.job ASC) AS job_titles,
    GROUP_CONCAT(DISTINCT e.season ORDER BY e.season ASC) AS season_numbers
FROM PawneeCommons.plays p
JOIN PawneeCommons.person pl ON p.pID = pl.pID
JOIN PawneeCommons.works w ON p.pID = w.pID
LEFT JOIN PawneeCommons.episode e ON w.eID = e.eID
GROUP BY pl.pID, pl.pFirstName, pl.pLastName
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	fil
1	SIMPLE	p	NULL	index	pID	pID	5	NULL	943	
1	SIMPLE	pl	NULL	eq_ref	PRIMARY	PRIMARY	4	pawneecommons.p.pID	1	
1	SIMPLE	w	NULL	ref	pID	pID	5	pawneecommons.p.pID	3	
1	SIMPLE	e	NULL	eq_ref	PRIMARY	PRIMARY	26	pawneecommons.w.eID	1	

Level	Code	Message
Note	1003	/* select#1 elect \`pawneecommons\`.\`pl\`.\`pID\` AS \`pID\`,\`pawneecommons\`.\`pl\`.\`pFirstName\` AS \`pFirstName\`,\`pawneecommons\`....


JOIN
Episode
JOIN
Works
JOIN
Plays
Person