```
USE BlogApplication;

# Take a look at what's in our tables
SELECT * FROM BlogPosts;
SELECT * FROM BlogComments;


# UNION DISTINCT
SELECT PostId, Title FROM BlogPosts WHERE PostId < 3
UNION DISTINCT
SELECT PostId, Title FROM BlogPosts WHERE PostId >= 2;
# UNION ALL
SELECT PostId, Title FROM BlogPosts WHERE PostId < 3
UNION ALL
SELECT PostId, Title FROM BlogPosts WHERE PostId >= 2;


# INTERSECTION (fully-qualified column names)
SELECT BlogPosts.PostId, BlogPosts.Title,
   BlogComments.CommentId, BlogComments.Content, BlogComments.PostId
FROM BlogPosts INNER JOIN BlogComments
        on BlogPosts.PostId = BlogComments.PostId;
# INTERSECTION (implied column names)
SELECT BlogPosts.PostId,Title,
   CommentId, BlogComments.Content, BlogComments.PostId
FROM BlogPosts INNER JOIN BlogComments
        on BlogPosts.PostId = BlogComments.PostId;
# INTERSECTION (* all columns short-hand)
SELECT *
FROM BlogPosts INNER JOIN BlogComments
        on BlogPosts.PostId = BlogComments.PostId;


# LEFT OUTER JOIN
# Default: difference of BlogPosts\BlogComments plus intersection
SELECT *
FROM BlogPosts LEFT OUTER JOIN BlogComments
        ON BlogPosts.PostId = BlogComments.PostId;
# LEFT OUTER JOIN
# Difference only
SELECT *
FROM BlogPosts LEFT OUTER JOIN BlogComments
        ON BlogPosts.PostId = BlogComments.PostId
WHERE BlogComments.PostId IS NULL;
# LEFT OUTER JOIN
# Anti-pattern: same as intersection
SELECT *
FROM BlogPosts LEFT OUTER JOIN BlogComments
        ON BlogPosts.PostId = BlogComments.PostId
```

```
WHERE BlogComments.PostId IS NOT NULL;
SELECT *
FROM BlogPosts INNER JOIN BlogComments
        on BlogPosts.PostId = BlogComments.PostId;


# CROSS JOIN
SELECT *
FROM BlogPosts CROSS JOIN BlogComments;
```