Example bucket:

An ETL with AWS Firehose writes a Parquet file to S3 every 15 minutes since 2019-01-01

```
mybucket/myprefix/date=2019-01-01/file001.parquet
…
mybucket/myprefix/date=2019-01-01/file096.parquet
…
mybucket/myprefix/date=2021-08-27/file096.parquet
```

The bucket now (2021-08) contains ~100k files.

Parquet columns

0: timestamp
1: sales
2: customer_id
3: vendor_id

Query

We want to compute some statistics over the last month. For example:

SELECT SUM(sales)
FROM sales_table(uri=s3://mybucket/myprefix)
WHERE date>=2021-08-01 and customer_id=1

Pushdown

Thanks to predicated and projection pushdown datafusion calls:

```
ParquetExec::try_from_path(
      “s3://mybucket/myprefix”,
      Some(Vec![1,2]),
      date>=2021-08-01 and customer_id=1,
      x,
      Y,
```

```
        None,
)
```

Which means that the ObjectStore can be called with:

```
S3ObjectStore.list("mybucket/myprefix", &[date>=2021-08-01, customer_id=1]
```

The S3 object store first emits a delimiter query which returns:

```
date=2019-01-01/
…
date=2021-08-27/
```

By parsing these folder names, it can deduce that the data is partitioned by a partitioning column called "date". it takes the expression with "date" inside, namely `date>=2021-08-01` and ignores the other expressions. Note that to avoid deducing the partition pattern, we could specify some metadata in the URI, e.g `s3://mybucket/myprefix?partition=date.`

It now knows that it only needs to list:

```
date=2021-08-01/
…
date=2021-08-27/
```

It can decide a strategy to do that for example:

- use prefix 2021-08-
- list [2021-08-01,...2021-08-27] one by one
- ...

Current issues

Currently, `ParquetExec::try_from_path` is called in ParquetTable::try_new without any predicate, to get the statistics and the schema. This means that the pushdown described above is efficient only if we use a table with known schema and we set `collect_statistics=false`.