1. Data models have been described in a separate figure.
2. MongoDB will be the persistence layer of choice for this problem. The choice of this database was primarily driven by two constraints of the application:
    a) Millions read per second
    b) A write frequency of 1 TB per hour

    Owing to no transactions and locking, MongoDB has a lower latency per query and hence can support a greater read and write frequencies than any relational databases. Also, the data that we would present to the user (given the description) does not warrant joins, hence MongoDB would again be a natural choice. Furthermore, given the updates, it would be nice to think about the scalability of the database. It is easier to shard data in MongoDB.

3. ETL have been discussed in producer.js and consumer.js
4. Average number of recipes updated per hour:

```
db.recipes.aggregate(
    [
        {
            "$group" : {
                "_id" : {
                    "recipe" : "$recipe_id",
                    "hour" : {
                        "$hour" : "$updated_date"
                    }
                },
                "num_recipes" : {
                    "$sum" : 1.0
                }
            }
        },
        {
            "$group" : {
                "_id" : null,
                "total" : {
                    "$sum" : "$num_recipes"
                },
                "recipe_count" : {
                    "$push" : "$num_recipes"
                }
            }
        },
        {
            "$project" : {
                "_id" : 0.0,
                "total" : 1.0,
                "average" : {
                    "$map" : {
                        "input" : "$recipe_count",
                        "as" : "item",
                        "in" : {
                            "$divide" : [
                                "$$item",
                                "$total"
```

```
                        ]
                    }
                }
            }
        }
    ],
    {
        "allowDiskUse" : false
    }
);
```

5. Number of recipes that got updated at 10.00 clock every year:

```
db.recipes.aggregate(
    [
        {
            "$project" : {
                "hour" : {
                    "$hour" : "$updated_date"
                },
                "recipe_name" : 1.0,
                "_id" : 0.0
            }
        },
        {
            "$match" : {
                "hour" : 10.0
            }
        },
        {
            "$group" : {
                "_id" : "$hour",
                "num_recipes" : {
                    "$sum" : 1.0
                }
            }
        }
    ],
    {
        "allowDiskUse" : false
    }
);
```