**Constraints:**

1. In **Dynamodb**, a **record** can be **uniquely** identified by a **primary key**.
2. The **primary key** can be a combination of two keys, namely, **PARTITION KEY (PK)** or **HASHKEY(HK)** and a **SORT KEY(SK)** or **RANGEKEY(RK).**
3. Multiple number of attributes can be stored along with the data but these attributes are non-indexable. A query involving these attributes will result in scanning the entire database and will incur more cost and time.
4. To facilitate a large number of **access patterns**, a piece of data is always stored using a combination of **PK** and **SK**.
5. Global Secondary indexes can be created to further facilitate more access patterns.
6. **Dynamodb transactions** allow us to create an **ACID** transaction consisting upto **25 atomic transactions**.
7. Transactions are **strongly consistent** and **cost double** the eventually consistent queries.

**Proposed scheme for storing ORDERS data**

When an order is received (even before payment), multiple records are stored in the database using the Transactions API. Let the order consist of products p1, p2, p3, …., pn from vendors v1, v2, v3, …., vn in quantities q1, q2, q3, …, qn, with marked price, m1, m2, m3, …. mn, and selling price, s1, s2, s3, …, sn.

Let the order is created by a user with id user\_id (phone number) at time time\_stamp, from location loc\_name.

The following entries will be made in the database.

**TRANSACT (1)**

1. Main order record

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PK | SK | Attribute 1 | Attribute 2 | Attribute 3 | Payment | Delivery | Order status | Refund | Net payment received | NET GST | NET commission | Order type |
| #ORDER#LOCATION<loc\_name> | Time\_stamp | Order\_id | User\_id | [ products] | {amount, type, status}  Type = cod | prepaid  Status = paid | pending | failed | {charge, address} | pending | out | delivered | rejected| failed | customer cancelled | admin cancelled | {amount, status}  Status= None | initiated | refunded | amount |  |  | Standard | express |

**Access patterns facilized**

1. Find all the orders in a particular location
2. Find all orders in a particular location in sorted manner wrt Timestamp
3. Find all orders in a particular location in sorted manner wrt Timestamp and in between Timestamp T1 and T2
4. Find all orders in a particular location with timestamp T (only one result ideally)
5. Reference record. It will be used in conjunction with transaction (2) to serve buyers

|  |  |  |  |
| --- | --- | --- | --- |
| PK | SK | ATTRIBUTE 1 | Attribute 2 |
| #ORDER#USER#<user\_id> | Time\_stamp | Order\_id | location |

**Access patterns facilized**

1. Find all the order ids of a user
2. Find all the order ids of a user in a sorted manner wrt timestamp
3. Find order id of an order created by a user at particular timestamp
4. Reference record. To index orders wrt order status

|  |  |  |
| --- | --- | --- |
| PK | SK | ATTRIBUTE 1 |
| ORDER#LOCATION#ORDERSTATUS# <loc\_name># <status> | timestamp | Order\_id |

**Access patterns facilized**

1. Browse orders by providing an order status in a sorted manner

**Deleted after successful delivery || Deleted after 1 year**

1. Reference record. To index orders wrt delivery status

|  |  |  |
| --- | --- | --- |
| PK | SK | ATTRIBUTE 1 |
| ORDER#LOCATION#DELIVERYSTATUS# <loc\_name>#<status> | timestamp | Order\_id |

**Access patterns facilized**

1. Browse orders by providing a delivery status in a sorted manner

**Deleted after successful delivery || Deleted after 1 year**

1. Reference record. To index orders wrt payment status

|  |  |  |
| --- | --- | --- |
| PK | SK | ATTRIBUTE 1 |
| ORDER#LOCATION# PAYMENTSTATUS#<loc\_name># <status> | timestamp | Order\_id |

**Access patterns facilized**

1. Browse orders by providing a payment status in a sorted manner

**Deleted after successful delivery || Deleted after 1 year**

1. Reference record. To index orders wrt refund status

|  |  |  |
| --- | --- | --- |
| PK | SK | ATTRIBUTE 1 |
| ORDER#LOCATION# REFUNDSTATUS#<loc\_name># <status> | timestamp | Order\_id |

**Access patterns facilized**

1. Browse orders by providing a refund status in a sorted manner

**Deleted after successful delivery || Deleted after 1 year**

1. Reference record. To keep track of express orders

|  |  |  |
| --- | --- | --- |
| PK | SK | ATTRIBUTE 1 |
| ORDER#LOCATION#EXPRESS#<loc\_name> | timestamp | Order\_id |

**Access patterns facilized**

1. Check if express order is placed in a given time range

**Deleted after successful delivery || Deleted after 1 year**

For each product P (p1-pn) in the order, a transaction is executed, which refers to a sub-order of a primary order.

**For i=1,2,3, …., n, that is, for all the products in an order, TRANSACT (2)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PK | SK | ATTR 1 | ATTR 2 | ATTR 3 | ATTR 4 | ATTR 5 | ATTR 6 | ATTR 7 | ATTR 8 | ATTR 9 | ATTR 10 |
| SUB\_ORDER#LOCATION#<order\_id> | <Vendor id (vi)><suborder\_sl\_no> | Unique Product Descriptor (pi) | Quantity (q1) | Marked price (m1) | Selling price (s1) | Time\_stamp | Order status:  Pending | vendor accepted | vendor rejected | ready | out | delivered | admin cancelled | failed | GST percentage | Commission percentage | Delivery charge | TCS |

**Access patterns facilized**

1. Given an order id, its sub orders can be traced and filtered with respect to vendor id

**For i=1,2,3, …., m, that is, for all the unique vendors, TRANSACT (3)**

|  |  |  |
| --- | --- | --- |
| PK | SK | ATTR 1 |
| ORDER#VENDOR#<vendor\_id> | timestamp | Order\_id |

**Access patterns facilized**

1. Search orders, given a vendor id.
2. Search orders, given a vendor id, sort wrt timestamp