

# CQL

```
CREATE KEYSPACE tickets WITH replication = {'class': 'SimpleStrategy',
'replication_factor': '3'} AND durable_writes = true;
use tickets;
CREATE TYPE OptionDetail (
    OptionID bigint,
    OptionOrder int,
    OptionName text,
    GroupType int,
    BetStage int,
    BetBar text,
    GameType int,
    MatchName text,
    MatchStartTime bigint,
    TournamentName text,
    TournamentID bigint,
    TourLevel int,
    SportID int,
    BillTime bigint,
    BetOption text,
    InsHomeScore int,
    InsAwayScore int,
    Odds double,
    OddsView text,
    ViewFmt int,
);

Create table matchTickets(MatchID bigint,
    ProductID int,
    BetType int,
    CompanyID int,
    TicketID bigint,
    TicketTime bigint,
    ConfirmedTime bigint,
    CompanyCustomerID
text,
    NBCustomerID bigint,
    BetAmount double,
    BetCurrency int,
    CurrencyRate double,
    BetSystemAmount double,
    MasterTicketID bigint,
    CustomerWin double,
    CustomerRisk double,
    Profit double,
    LastCancelTime bigint,
    TicketStatus int,
    RunningStage int,
    MatchResult text,
    Option Frozen<OptionDetail>,
```

```

                                OptionResult int,
                                Primary key((ProductID, CompanyID),
CompanyCustomerID, TicketTime,TicketID ))
WITH CLUSTERING ORDER BY (companycustomerid ASC, tickettime DESC, ticketid
DESC);
CREATE INDEX idx_matchtickets_matchid ON tickets.matchtickets (matchid);
Create index idx_matchtickets_ticketid on tickets.matchtickets(ticketid);
CREATE INDEX idx_matchtickets_ticketstatus ON tickets.matchtickets
(ticketstatus);
Create index idx_matchtickets_nbCustomerId on tickets.matchtickets
(nbcustomerid);

Create Type SubCombo(
    SubComboID bigint,
    Profit double,
    OptionIds Set<bigint>
);

Create Type ComboBet (
    ComboID bigint,
    Num int,
    Cnt int,
    BetAmount double,
    ComboProfit double,
    SubCombos set<frozen<SubCombo>>
);

Create Type ComboOption(
                                TicketId bigint,
                                OptionResult int,
                                Profit double,
                                MatchID bigint,
                                OptionID bigint,
                                OptionOrder int,
                                OptionName text,
                                GroupType int,
                                BetStage int,
                                BetBar text,
                                GameType int,
                                MatchName text,
                                MatchStartTime bigint,
                                TournamentName text,
                                TournamentID bigint,
                                TourLevel int,
                                SportID int,
                                BillTime bigint,
                                BetOption text,
                                InsHomeScore int,
                                InsAwayScore int,
                                Odds double,
                                OddsView text,
                                ViewFmt int,
                                RunningStage int
);

```

```

Create Table MasterComboTickets(MasterTicketID bigint,
                                TicketStatus int,
                                TicketTime bigint,
                                BillTime
                                bigint,
                                CompanyID int,
                                ProductID int,
                                CompanyCustomerID
                                text,
                                NBCustomerID bigint,
                                Currency int,
                                CurrencyRate double,
                                TotalAmount double,
                                TotalProfit double,
                                ComboBets Map<bigint, Frozen<ComboBet>>,
                                TicketOptions Map<bigint,
                                Frozen<ComboOption>>,
                                UnBilledTicketOptions Set<bigint>,
                                Primary Key(MasterTicketID));
Create table CompanyCustomers(CompanyID int
                              , CompanyCustomerID text
                              , NBCustomerID bigint
                              , CustomerToken text
                              , CreateTime bigint
                              , Primary key (CompanyID, CompanyCustomerID));
CREATE INDEX idx_MasterCombo_ticketStatus ON tickets.MasterComboTickets
(TicketStatus);
create index idx_customers_token on CompanyCustomers(customerToken);
Create table MatchsTime(MatchID bigint, MatchStartTime bigint, Primary Key
(MatchID));

Create MATERIALIZED VIEW TicketsByNBUserID as select * from matchtickets
where CompanyCustomerID is not null and CompanyId is not null and
ProductId is not null and TicketTime is not null and ticketid is not null
and NBCustomerID is not null Primary key(NBCustomerID,TicketTime,
productid, companyid, companycustomerid, ticketid);

CREATE MATERIALIZED VIEW tickets.ViewForTimeOrderQuery AS
SELECT *
FROM tickets.matchtickets
WHERE companyid IS NOT null AND productid IS NOT null AND tickettime
IS NOT null AND ticketstatus IS NOT null and companycustomerid is not null
and ticketid is not null
PRIMARY KEY ((productid, companyid), tickettime, ticketid,
ticketstatus, companycustomerid)
WITH CLUSTERING ORDER BY (tickettime DESC) ;

CREATE MATERIALIZED VIEW tickets.ViewForAmountOrderQuery AS
SELECT *
FROM tickets.matchtickets
WHERE companyid IS NOT null AND productid IS NOT null AND tickettime

```

IS NOT null and betamount is not null AND companycustomerid is not null  
and ticketid is not null

PRIMARY KEY ((productid, companyid),betamount, tickettime,  
companycustomerid, ticketid)

WITH CLUSTERING ORDER BY (betamount DESC, tickettime DESC) ;

Create Table MatchResults

(MatchId bigint Primary Key,

Results text,

lastBillMsgTime text

);