**3NF Schema:**

*1. Table Definitions:*

CREATE TABLE `r1\_alt` (

`company` int NOT NULL,

`event` int NOT NULL,

`time` int NOT NULL,

PRIMARY KEY (`event`,`time`),

UNIQUE KEY `idx\_r1\_alt\_unique\_company\_time` (`company`,`time`)

) ENGINE=InnoDB DEFAULT CHARSET=ascii COLLATE=ascii\_bin;

CREATE TABLE `r2` (

`event` int NOT NULL,

`venue` int NOT NULL,

`time` int NOT NULL,

PRIMARY KEY (`event`,`time`),

UNIQUE KEY `idx\_r2\_unique\_venue\_time` (`venue`,`time`)

) ENGINE=InnoDB DEFAULT CHARSET=ascii COLLATE=ascii\_bin

*2. Trigger to enforce FD event -> company:*

CREATE DEFINER=`root`@`localhost` TRIGGER `R1\_alt\_BEFORE\_INSERT` BEFORE INSERT ON `r1\_alt` FOR EACH ROW BEGIN

IF @TRIGGER\_DISABLED = 0 THEN

set @exist = IF(EXISTS(SELECT \* FROM R1\_alt WHERE R1\_alt.event = new.event and R1\_alt.company<>new.company),1,0);

if @exist = 1 THEN

insert into R1\_alt VALUES(new.company,new.event,new.time) ;

end if;

END IF;

END

*3. Stored Procedures for Inserts:*

CREATE DEFINER=`root`@`localhost` PROCEDURE `populate\_ex\_3nf\_R1\_alt`(IN numberOfRows INT)

BEGIN

DECLARE counter INT;

SET counter = 0;

WHILE (counter < numberOfRows) DO

INSERT INTO R1\_alt(company,event,time) VALUES(0+counter\*6,0+counter\*6,0+counter\*6);

INSERT INTO R1\_alt(company,event,time) VALUES(0+counter\*6,1+counter\*6,1+counter\*6);

INSERT INTO R1\_alt(company,event,time) VALUES(2+counter\*6,2+counter\*6,2+counter\*6);

INSERT INTO R1\_alt(company,event,time) VALUES(2+counter\*6,2+counter\*6,3+counter\*6);

INSERT INTO R1\_alt(company,event,time) VALUES(2+counter\*6,2+counter\*6,4+counter\*6);

INSERT INTO R1\_alt(company,event,time) VALUES(5+counter\*6,5+counter\*6,4+counter\*6);

SET counter = counter + 1;

END WHILE;

END

CREATE DEFINER=`root`@`localhost` PROCEDURE `populate\_ex\_3nf\_R2`(IN numberOfRows INT)

BEGIN

DECLARE counter INT;

SET counter = 0;

WHILE (counter < numberOfRows) DO

INSERT INTO R2(event,venue,time) VALUES(0+counter\*6,0+counter\*6,0+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(1+counter\*6,0+counter\*6,1+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(2+counter\*6,0+counter\*6,2+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(2+counter\*6,0+counter\*6,3+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(2+counter\*6,4+counter\*6,4+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(5+counter\*6,5+counter\*6,4+counter\*6);

SET counter = counter + 1;

END WHILE;

END

*4. Stored Procedures for Updates, illustrated on update of events based on times:*

CREATE DEFINER=`root`@`localhost` PROCEDURE `update-event-based-on-time-and-company`() BEGIN

start transaction;

UPDATE R1\_alt, R2

SET R1\_alt.event = R1\_alt.event + 1, R2.event = R2.event + 1

WHERE R1\_alt.event=R2.event and R1\_alt.time=R2.time and R1\_alt.time mod 6=4 and R1\_alt.company mod 6=2;

if EXISTS(SELECT R1\_alt.event FROM R1\_alt GROUP BY R1\_alt.event HAVING COUNT(DISTINCT R1\_alt.company)>1) THEN

ROLLBACK;

ELSE

COMMIT;

end if;

END

**2-CONF Schema:**

*1. Table Definitions:*

CREATE TABLE `R1` (

`event` int NOT NULL,

`company` int NOT NULL,

PRIMARY KEY (`event`)

) ENGINE=InnoDB DEFAULT CHARSET=ascii COLLATE=ascii\_bin

CREATE TABLE `R2` (

`event` int NOT NULL,

`venue` int DEFAULT NULL,

`time` int NOT NULL,

PRIMARY KEY (`event`,`time`),

UNIQUE KEY `UNIQUE\_vt` (`venue`,`time`)

) ENGINE=InnoDB DEFAULT CHARSET=ascii COLLATE=ascii\_bin

CREATE TABLE `R3` (

`company` int NOT NULL,

`venue` int DEFAULT NULL,

`time` int NOT NULL,

PRIMARY KEY (`company`,`time`),

UNIQUE KEY `UNIQUE\_vt` (`venue`,`time`)

) ENGINE=InnoDB DEFAULT CHARSET=ascii COLLATE=ascii\_bin

*2. Stored Procedure for Insertions:*

CREATE DEFINER=`root`@`localhost` PROCEDURE `populate\_ex\_conf\_R1`(IN numberOfRows INT)

BEGIN

DECLARE counter INT;

SET counter = 0;

WHILE (counter < numberOfRows) DO

INSERT INTO R1(event,company) VALUES(0+counter\*6,0+counter\*6);

INSERT INTO R1(event,company) VALUES(1+counter\*6,0+counter\*6);

INSERT INTO R1(event,company) VALUES(2+counter\*6,2+counter\*6);

INSERT INTO R1(event,company) VALUES(5+counter\*6,5+counter\*6);

SET counter = counter + 1;

END WHILE;

END

CREATE DEFINER=`root`@`localhost` PROCEDURE `populate\_ex\_conf\_R2`(IN numberOfRows INT)

BEGIN

DECLARE counter INT;

SET counter = 0;

WHILE (counter < numberOfRows) DO

INSERT INTO R2(event,venue,time) VALUES(0+counter\*6,0+counter\*6,0+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(1+counter\*6,0+counter\*6,1+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(2+counter\*6,0+counter\*6,2+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(2+counter\*6,0+counter\*6,3+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(2+counter\*6,4+counter\*6,4+counter\*6);

INSERT INTO R2(event,venue,time) VALUES(5+counter\*6,5+counter\*6,4+counter\*6);

SET counter = counter + 1;

END WHILE;

END

CREATE DEFINER=`root`@`localhost` PROCEDURE `populate\_ex\_conf\_R3`(IN numberOfRows INT)

BEGIN

DECLARE counter INT;

SET counter = 0;

WHILE (counter < numberOfRows) DO

INSERT INTO R3(company,time,venue) VALUES(0+counter\*6,0+counter\*6,0+counter\*6);

INSERT INTO R3(company,time,venue) VALUES(0+counter\*6,1+counter\*6,0+counter\*6);

INSERT INTO R3(company,time,venue) VALUES(2+counter\*6,2+counter\*6,0+counter\*6);

INSERT INTO R3(company,time,venue) VALUES(2+counter\*6,3+counter\*6,0+counter\*6);

INSERT INTO R3(company,time,venue) VALUES(2+counter\*6,4+counter\*6,4+counter\*6);

INSERT INTO R3(company,time,venue) VALUES(5+counter\*6,4+counter\*6,5+counter\*6);

SET counter = counter + 1;

END WHILE;

END