A Relational Database for Fermilab magnet test facility

Purpose:

To ensure proper functioning of the particle accelerator, its each magnetic component has to be calibrated periodically, the need of this database is to record the behaviour of the magnetic component for specific input and to maintain the log of who is performing it, how and when the calibration is performed.

ER diagram for my database

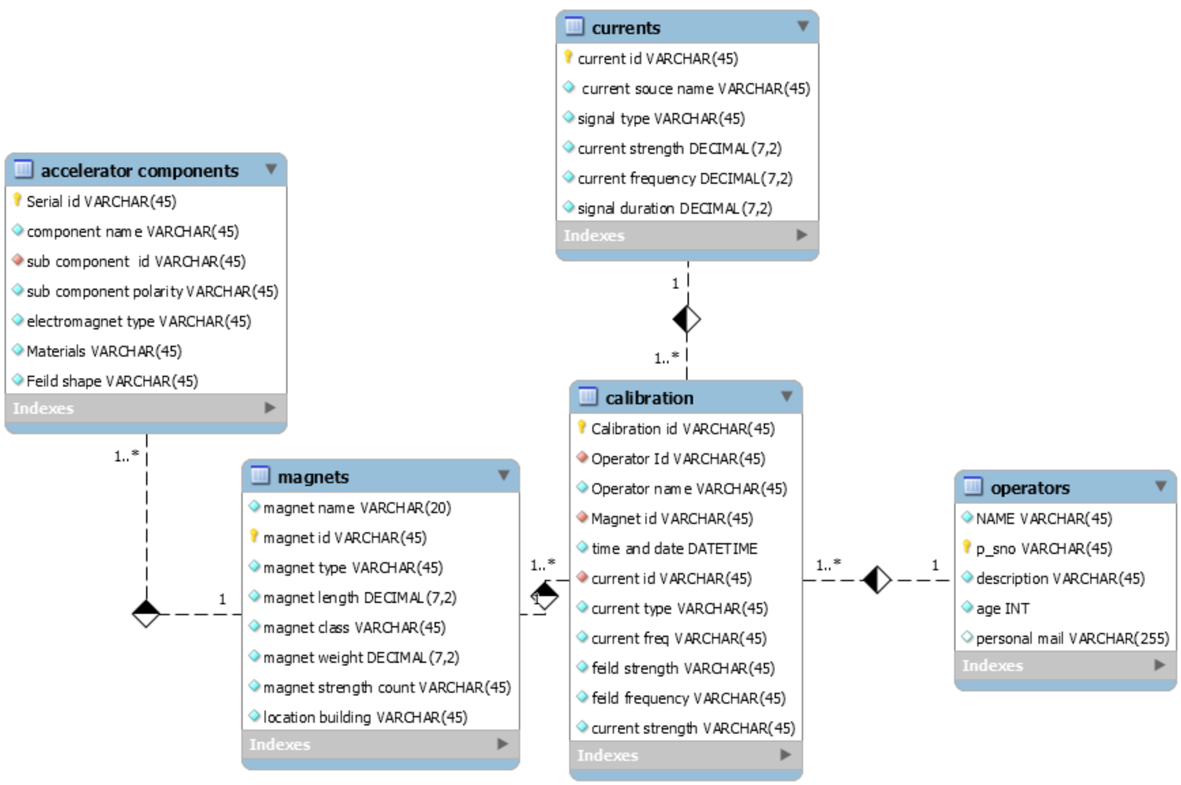


Table 1: Accelerator components

Accelerator is used to energize the particles to high speeds and making collisions and to study the collision for the created particles and rays. To stabilize and to control the particles magnetic components are used.

Accelerator components

Serial id	component name	sub component id	sub component polarity	electromagnet type	Materials	Feild shape
Z011	bending magnet	M1210	dipole	normallyconducting	Cu,Fe	axial
Z290	focusing magnet	M2167	quadrupole	superconducting	Nb,Fe,Al	axial
Z345	magnetic horn	M9999	quadrupole	normallyconducting	Ni,Al	toroidal
Z467	Undulator	M7676	dipole	permanent magnet	Va	axial
Z510	Solenoid magnet	M9879	quadrupole	superconducting	Fe,Al,Nb,Ti	axial
Z642	Booster kicker magnet	M2189	dipole	normallyconducting	Cu,Al,Fe	axial
Z703	Storage ring magnet	M9999	quadrupole	superconducting	Fe,Al,Nb,Ti	toroidal
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Each huge component of accelerator is made up of smaller powerful magnets like dipoles and quadrupoles, Type says about whether its naturally magnetic (permanent magnet) or normally conducting or superconducting (we have to power it), Material infers its composition and the field shape tells us whether the field exists along the axis or along the perimeter

The meaning of the first tuple is: "The gigantic bending magnet is made of subcomponent M1210 which is a dipole, It is a normally conducting electro-magnet, with field along its axis and made primarily of copper and iron"

Primary Key: Serial ID

Other Candidate Key: component name

1st normal form:

Materials is a multivalued attribute hence the table violates 1NF, to normalize it we have to make the attribute values atomic, so we can create a separate table with Serial id and Materials (here both combine to form the primary key) and the Serial id and remaining attributes in a separate table

Accelerator components A

Serial id	component name	sub component id	sub component polarity	electromagnet type	Feild shape
Z011	bending magnet	M1210	dipole	normallyconducting	axial
Z290	focusing magnet	M2167	quadrupole	superconducting	axial
Z345	magnetic horn	M9999	quadrupole	normallyconducting	toroidal
Z467	Undulator	M7676	dipole	permanent magnet	axial
Z510	Solenoid magnet	M9879	quadrupole	superconducting	axial
Z642	Booster kicker magnet	M2189	dipole	normallyconducting	axial
Z703	Storage ring magnet	M9999	quadrupole	superconducting	toroidal
NULL	NULL	NULL	NULL	NULL	NULL

Component Material

Serial_Id	Material
Z011	Cu
Z011	Fe
Z290	Al
Z290	Fe
Z290	Nb
Z345	Al
Z345	Ni
Z467	Va
Z510	Al
Z510	Fe
Z510	Nb
Z510	Ti
Z642	Al
Z642	Cu
Z642	Fe
Z703	Al
Z703	Fe
Z703	Nb
Z703	Ti
NULL	NULL

2nd normal form :

Since the candidate keys are of single attribute the table is automatically in 2nd normal form

3rd normal form :

Sub component id and polarity belongs to “magnets” table, there the primary key “id” determines the “polarity “ attribute, {sub_component_id} →{sub_component_polarity} is the functional dependency, it violates 3NF because {sub_component_id} is a non prime attribute, or there exists a transitive dependency from {Serial_Id} to {sub_component_polarity}. We can normalize by splitting the tables as follows where the “id” forms the primary key of “sub component details” table.

This can be done using the following query:

SQL QUERY: Create table sub_component_details as select distinct sub_component_id, sub_component_polarity from Accelerator_Components;

Alter table Accelerator_components drop sub_component_polarity;

Accelerator Components A1

Serial_id	component_name	sub_component_id	electromagnet_type	Feild_shape
Z011	bending magnet	M1210	normallyconducting	axial
Z290	focusing magnet	M2167	superconducting	axial
Z345	magnetic horn	M9999	normallyconducting	toroidal
Z467	Undulator	M7676	permanent magnet	axial
Z510	Solenoid magnet	M9879	superconducting	axial
Z642	Booster kicker magnet	M2189	normallyconducting	axial
Z703	Storage ring magnet	M9999	superconducting	toroidal

Sub Component Details

sub_component_id	sub_component_polarity
M1210	dipole
M2167	quadrupole
M2189	dipole
M7676	dipole
M9879	quadrupole
M9999	quadrupole

BCNF:

3 tables are in BCNF because for “Accelerator Components A1” we have 2 candidate keys both have a single attribute{Serial Id},{Component name} and remaining 2 tables have no functional dependencies.

4th and 5th normal form:

There are no multivalued and join dependencies in the tables hence they satisfy 4NF and 5NF.

Final 3 tables are “Accelerator Components A1”, “Sub Component Details”, ” Component Material”.

TABLE 2: Magnets : (Sub components)

Each accelerator component is made up of magnetic sub component such as dipoles and quadrupoles. There are around 150 quadrupoles and 240 dipoles in the Fermi lab, and are used in different building complexes: {A, B, C, D, E}.

Each subcomponent is named depending on which accelerator component it is used

There are two different kinds of dipole magnets. Their magnet type specifications are:

1)Type A - Length is 6.096m (240in.). Magnet count is 108 and magnet weight is 40,000lb.

2)Type B - Length is 6.096m (240in.). Magnet count is 108 and magnet weight is 40,000lb.

Magnets

magnet name	magnet id	magnet type	magnet length	magnet class	magnet weight	magnet strength count	location building
alpha11	M1210	Dipole	240.00	A	40000.00	108	A,B
beta23	M2167	Qualdrupole	84.00	R	8740.00	128	B,D
beta33	M2189	Dipole	160.00	B	27000.00	64	A
beta45	M3170	Quadrupole	100.00	Q	10300.00	32	B
alpha23	M3230	Dipole	240.00	A	40000.00	108	A,C
alpha34	M4303	Quadrupole	116.00	P	12000.00	48	D,E
gamma2	M7676	Dipole	160.00	B	27000.00	64	C
gamma3	M9824	Quadrupole	100.00	Q	10300.00	32	A,C
gamma3	M9879	Quadrupole	84.00	R	8740.00	128	A,B
staralpha	M9999	Quadrapole	116.00	P	12000.00	48	D,E

Quadrupole Magnets

There are three different kinds of quadrupole magnets. Their magnet type specifications are:

1) Type P- Length is 2.945m (116in.). Magnet count is 48 and magnet weight is 12,000lb.

2) Type Q- Length is 2.539m (100in.). Magnet count is 32 and magnet weight is 10,300lb.

3) Type R - Length is 2.134m (84in.). Magnet count is 128 and magnet weight is 8,740lb.

We can clearly see that the class determines the count, length and weight, Also each magnet type has a different class. This is the reason, the following 2 functional dependencies exist

{magnet_class} →{magnet_type}

{magnet_class}→{magnet_weight, magnet_length, magnet_strength_count}

The meaning of the first tuple is : “The magnetic subcomponent named “alpha11” with id “M1210” is a dipole of class A and used in the building complexes “A” and “B” “

From the magnet class and in which complex (place) it is used, we can tell its description

Primary Key: magnet id

Other candidate keys: {magnet_class, location_building}

1NF : Each dipole/quadrupole is used in many places, so “location_building” attribute is a multivalued, hence it violates 1NF, To satisfy 1NF lets create a separate table with {magnet id , location building} as the attributes, both combined forms the primary key and the remaining attributes in the other table

Magnets A

magnet name	magnet id	magnet type	magnet length	magnet class	magnet weight	magnet strength count
alpha11	M1210	Dipole	240.00	A	40000.00	108
beta23	M2167	Qualdrupole	84.00	R	8740.00	128
beta33	M2189	Dipole	160.00	B	27000.00	64
beta45	M3170	Quadrupole	100.00	Q	10300.00	32
alpha23	M3230	Dipole	240.00	A	40000.00	108
alpha34	M4303	Quadrupole	116.00	P	12000.00	48
gamma2	M7676	Dipole	160.00	B	27000.00	64
gamma3	M9824	Quadrupole	100.00	Q	10300.00	32
gamma3	M9879	Quadrupole	84.00	R	8740.00	128
staralpha	M9999	Quadrapole	116.00	P	12000.00	48

Magnet_Locations

magnet_id	complex name
M1210	A
M1210	B
M2167	B
M2167	D
M2189	A
M3170	B
M3230	A
M3230	C
M4303	D
M4303	E
M7676	C
M9824	A
M9824	C
M9879	A
M9879	B
M9999	D
M9999	E

2NF :

The primary key{magnet_id} is a single attribute, the candidate key {magnet class, location} is lost while table splitting, therefore the relation is in 2NF.

3NF :

A non prime attribute {magnet_class} functionally determines {magnet_type, magnet_length, magnet_weight, magnet_stregth_count}, therefore a transitive dependency from magnet_id to these attributes exsist, hence the table is not in 3NF. By creating a new table with "magnet class" as the prime attribute we can normalize the table. The remaining attributes are kept in the original table with "magnet_id" as the primary key. This can be done using the following queries

SQL QUERY:

Create table magnet_class as select distinct magnet_class, magnet_type , magnet_length, magnet_weight , magnet_strength_count from magnets;

and then droping these attibutes except "magnet_class" attribute in the original table.

Magnets_1_A

magnet_name	magnet_id	magnet_class
alpha11	M1210	A
beta23	M2167	R
beta33	M2189	B
beta45	M3170	Q
alpha23	M3230	A
alpha34	M4303	P
gamma2	M7676	B
gamma3	M9824	Q
gamma3	M9879	R
staralpha	M9999	P

Magnet_class

magnet type	magnet length	magnet class	magnet weight	magnet strength count
Qualdrupole	84.00	R	8740.00	128
Dipole	160.00	B	27000.00	64
Quadrupole	100.00	Q	10300.00	32
Dipole	240.00	A	40000.00	108
Quadrupole	116.00	P	12000.00	48

BCNF:

The 3 tables are in BCNF because there are only functional dependencies from the primary key “magnet_id” in Magnets_1_A table and “magnet_class” in Magnet_class table. Moreover there are no functional dependencies in magnet_location table.

4NF and 5NF:

Due to the absence of multivalued dependencies and join dependencies the 3 tables satisfy 4NF and 5NF respectively

The final 3 tables are “Magnet_Class”, “Magnet_1_A”, “Magnet_locations”

Table 3: Calibration

Each of the magnetic subcomponent “ dipoles and quadrupoles” will be calibrated periodically by an operator, This table act as a “log book” telling us about the calibration process.

Calibration

Calibration id	Operator Id	Operator name	Magnet id	time and date	current id	current type	current freq	feild strength	feild frequency	current strength
C1234	10101	Daniel Robert	M1210	2015-11-05 14:29:30	I212	square	53	20	106	990
C2134	10987	Harry bond	M3230	2015-07-04 10:02:03	I212	square	53	20	106	990
C3223	19876	A D Rushell	M9999	2015-06-03 01:20:32	I313	sinusoid	106	40	212	500
C4214	14325	H P Waltm...	M3170	2015-11-09 12:02:21	I413	sinusoid	53	40	106	200
C4312	10990	Rams Junior	M4303	2015-06-12 12:03:34	I888	triangular	5	60	10	600
C5432	10987	Harry bond	M7676	2015-08-04 14:21:40	I313	sinusoid	106	40	212	200
C5473	10890	H D Glass	M9999	2015-08-05 07:12:36	I413	sinusoid	53	40	106	200
C5678	10767	B C Brown	M2167	2015-04-05 09:25:25	I888	triangular	5	60	10	600
C7123	10345	C S Mishra	M9879	2015-05-08 11:22:34	I413	sinusoid	53	40	106	200
C8901	10101	Daniel Robert	M1210	2015-10-07 14:29:36	I313	sinusoid	106	40	212	500
C9087	10767	B C Brown	M2189	2016-12-30 10:28:36	I512	square	106	60	212	990

Each subcomponent is taken to the calibration room where it is calibrated by a professional operator. Each calibration is given a calibration ID. The operator id, name, time and date of calibration are noted. What kind of currents are given as input and what kind of fields are observed are also noted.

The currents will be discussed in currents table, the ouput field strength is in milli Tesla(10^-3 Tesla) , and both frequencies are in Mhz (10^6 hz).

These magnetic subcomponents are calibrated using a specific input currents, each current signal is given a name/id , Each current produces a unique magnetic field output which is measured. Field frequency depends on current frequency and strength depends upon the current and the magnet.

The folowing functional dependencies exsist

Current_id →{current_type, current_freq, current_strength}

Current_freq→field_frequency

{Current_id,Magnet,id}→field_strength

Operator id itself contains the information about his name

Operator_Id→Operator_name

Primary Key : Calibration ID

Other candidate Key:{ Operator_ID, Magnet_Id, time_and_date, current_ID,}

1NF:

There are no multi valued attributes and composite attributes present in this table so it satisfies 1NF

2NF:

The non prime attribute "Operator_name" is not fully functionally dependent on the Candidate Key ,but determined by the part of it i.e "Operator_Id", this is a partial dependency hence it violates 2NF.

It can be normalized by creating a new table as

SQL QUERY: Create table calib_operator as select distinct operator_id, opertor-name from calibration;

Alter table calibration drop operator_name;

And then making "ID" as the primary key of the calib_operator table.

Calibration_A

Calibration_id	Operator_id	Magnet_id	time_and_date	current_id	current_type	current_freq	feild_strength	feild_freq	current_strength
C1234	10101	M1210	2015-11-05 14:29...	I212	square	53	20	106	990
C2134	10987	M3230	2015-07-04 10:02...	I212	square	53	20	106	990
C3223	19876	M9999	2015-06-03 01:20...	I313	sinusoid	106	40	212	500
C4214	14325	M3170	2015-11-09 12:02...	I413	sinusoid	53	40	106	200
C4312	10990	M4303	2015-06-12 12:03...	I888	triangular	5	60	10	600
C5432	10987	M7676	2015-08-04 14:21...	I313	sinusoid	106	40	212	200
C5473	10890	M9999	2015-08-05 07:12...	I413	sinusoid	53	40	106	200
C5678	10767	M2167	2015-04-05 09:25...	I888	triangular	5	60	10	600
C7123	10345	M9879	2015-05-08 11:22...	I413	sinusoid	53	40	106	200
C8901	10101	M1210	2015-10-07 14:29...	I313	sinusoid	106	40	212	500
C9087	10767	M2189	2016-12-30 10:28...	I512	square	106	60	212	990

Calib_Operator

operator_Id	operator_name
10101	Daniel Robert
10345	C S Mishra
10767	B C Brown
10890	H D Glass
10987	Harry bond
10990	Rams Junior
14325	H P Waltmann
19876	A D Rushell

One more partial dependency {Current_id}→{current_freq, current_strength, current_type} from the prime attribute 'Current_id" violates 2NF but normailizing now, destroys the functional dependency {current_freq}→{field-freq}

To preserve this FD lets check for 3NF.

3NF:

The fuctional dependency {current_freq}→{field-freq} violates 3NF because there is a transitive dependency from primary key Calibration_ID to field_freq through current_freq;

It can be normailized by creating a new table for these 2 attributes

SQL QUERY: Create table Output_freq as select distinct current_freq,field_freq from Calibration;

Alter table calibration drop feild_freq;

Then setting current_freq as the primary key of the new table

Calibration_A1

Calibration_id	Operator_id	Magnet_id	time_and_date	current_id	current_type	current_freq	feild_strength	current_strength
C1234	10101	M1210	2015-11-05 14:29...	I212	square	53	20	990
C2134	10987	M3230	2015-07-04 10:02...	I212	square	53	20	990
C3223	19876	M9999	2015-06-03 01:20...	I313	sinusoid	106	40	500
C4214	14325	M3170	2015-11-09 12:02...	I413	sinusoid	53	40	200
C4312	10990	M4303	2015-06-12 12:03...	I888	triangular	5	60	600
C5432	10987	M7676	2015-08-04 14:21...	I313	sinusoid	106	40	200
C5473	10890	M9999	2015-08-05 07:12...	I413	sinusoid	53	40	200
C5678	10767	M2167	2015-04-05 09:25...	I888	triangular	5	60	600
C7123	10345	M9879	2015-05-08 11:22...	I413	sinusoid	53	40	200
C8901	10101	M1210	2015-10-07 14:29...	I313	sinusoid	106	40	500
C9087	10767	M2189	2016-12-30 10:28...	I512	square	106	60	990

Output_freq

current_freq	feild_frequency
106	212
5	10
53	106

Now we can normalize it to 2NF by creating a separate table for the partial dependency {Current_id} → {current_freq, current_strength, current_type} ,.then setting "current_id" as the primary key for the new table

SQL QUERY: Create table Calib_currents as select distinct current-id, current_type, current_freq,

current_strength from calibration;

Then dropping the non prime attributes from the original table.

The original foreign key constraint from attribute "current_freq" of "output frequency" table to "current_freq" of "calibration" table is removed and new constraint is created to "current_freq" of "calib_currents" table

Calibration_A2

Calibration_id	Operator_id	Magnet_id	time_and_date	current_id	feild_strength
C1234	10101	M1210	2015-11-05 14:29...	I212	20
C2134	10987	M3230	2015-07-04 10:02...	I212	20
C3223	19876	M9999	2015-06-03 01:20...	I313	40
C4214	14325	M3170	2015-11-09 12:02...	I413	40
C4312	10990	M4303	2015-06-12 12:03...	I888	60
C5432	10987	M7676	2015-08-04 14:21...	I313	40
C5473	10890	M9999	2015-08-05 07:12...	I413	40
C5678	10767	M2167	2015-04-05 09:25...	I888	60
C7123	10345	M9879	2015-05-08 11:22...	I413	40
C8901	10101	M1210	2015-10-07 14:29...	I313	40
C9087	10767	M2189	2016-12-30 10:28...	I512	60

Calib_currents

current_id	current_type	current_freq	current_strength
I212	square	53	990
I313	sinusoid	106	500
I413	sinusoid	53	200
I888	triangular	5	600
I512	square	106	990

The FD : {Current_id,Magnet,id}→field_strength. violates 2NF because the subset of the candidate key is determining a non prime attrinbute

To normalize into 2NF we can create a separate table for this partial dependency with operator id and magnet id as the primary key, then we can remove the field strength attribute from the original table.

SQL QUERY: Create table output_strength as select distinct current_id, magnet_id, feild_strength from calibration;

Alter table calibration drop field_strength;

Output_Strength

output_current_id	tested_magnet_id	feild_strength
I212	M1210	20
I212	M3230	20
I313	M1210	40
I313	M7676	40
I313	M9999	40
I413	M3170	40
I413	M9879	40
I413	M9999	40
I512	M2189	60
I888	M2167	60
I888	M4303	60

Calibration_A3

Calibration_id	Operator_id	Magnet_id	time_and_date	current_id
C1234	10101	M1210	2015-11-05 14:29...	I212
C2134	10987	M3230	2015-07-04 10:02...	I212
C3223	19876	M9999	2015-06-03 01:20...	I313
C4214	14325	M3170	2015-11-09 12:02...	I413
C4312	10990	M4303	2015-06-12 12:03...	I888
C5432	10987	M7676	2015-08-04 14:21...	I313
C5473	10890	M9999	2015-08-05 07:12...	I413
C5678	10767	M2167	2015-04-05 09:25...	I888
C7123	10345	M9879	2015-05-08 11:22...	I413
C8901	10101	M1210	2015-10-07 14:29...	I313
C9087	10767	M2189	2016-12-30 10:28...	I512

Now the tables satisfies 2NF and 3NF.

BCNF:

There are no non trivial FD s in any of these resultant 5 tables hence BCNF is satisfied by them

4NF and 5NF

There are no MVD and join dependencies hence the 5 tables are in 4NF and 5NF.

Resultant tables after normalization: “Calibration_A3”, “Output_Strength”, “Output_freq”, “Calib_currents”, “Calib_operator”.

TABLE 4: CURRENTS

Since each component are sophisticated instrumnets they are calibrated using a special kind of standard current signals, There are special instruments to give these standard current signals, Each current signal is a pulse of specific duration. This table tells us about their type, frequency, strength, duration

Currents

current id	current souce name	signal type	current strength	current frequency	signal duration
I212	F17 Kicker	square	990.00	53.00	200.00
I313	E48 Kicker	sinusoid	500.00	106.00	400.00
I413	E48 Kicker	sinusoid	200.00	53.00	400.00
I512	F17 Kicker	square	990.00	106.00	200.00
I777	FO Lambertson	triangular	300.00	20.00	600.00
I888	FO Lambertson	triangular	600.00	5.00	600.00

The meaning of the first tuple is : "The current source "F17 Kicker" is used to produce a square signal of strength-990 Amphere, 53 Mhz frequency and duration 200 milli seconds and its given an id called I212"

Source determines the signal type and signal duration

FD: {current_source_name}→{signal_type,signal_duration}

Primary Key: current_id

Candidate Key: {current_source_name, current_strength, current_frequency}

<u>1NF:</u>

Since there are no multi valued attributes and composite attributes present in this table , it satisfies 1NF .

<u>2NF</u>:

The proper subset of the candidate key { current_source_name} determines a non prime attribute by itself alone, So this is a partial dependency hence 2NF is violated

Lets create a separate table for this FD to normalize the relation,

SQL QUERY: Create table current_mode_relation as select distinct current_source_name, signal_type,

signal_duration from currents;

Then make current_source_name as the primary key of the new table

And then droping these non prime attributes here from the original table

Currents_A

current_id	current_source_name	current_strength	current_frequency
I212	F17 Kicker	990.00	53.00
I313	E48 Kicker	500.00	106.00
I413	E48 Kicker	200.00	53.00
I512	F17 Kicker	990.00	106.00
I777	FO Lambertson	300.00	20.00
I888	FO Lambertson	600.00	5.00

Current_mode_relation

current_source_name	signal_type	signal_duration
E48 Kicker	sinusoid	400.00
F17 Kicker	square	200.00
FO Lambertson	triangular	600.00

<u>3NF and BCNF</u> :

There are only trivial FD s in these 2 tables hence they are in 3NF and BCNF

<u>4NF and 5NF</u> :

There are no MVDs and join dependencies hence they are in 4NF and 5NF respectively.

Table 5: Operators:

These are the descriptions of the important personnel working in the Fermilab Magnet System department. These people only calibrate the individual subcomponents ”dipoles” and “quadrupoles” in the facility, Their description ,name, age and contact details are stored in this relation. Each operator is given a unique id “p_sno” using which we can identify and tell about them.

The meaning of the first tuple is : “Daniel Robert is a 49 year old mechanical technician with id 10101 and his contact mail is drobert12@gmail.com”

NAME	p_sno	description	age	personal mail
Daniel robert	10101	mechanical technician	49	drobert12@gmail.com
C S Mishra	10345	mechanical technician	56	mishra17@gmail.com
B C Brown	10767	instrument machinist	47	brownbc@gmail.com
H D Glass	10890	leading Scientist	43	hdglass13@gmail.com
D J Harding	10980	leading Scientist	45	hardingdj@gmail.com
Harry bond	10987	engineering physicist	45	harrybond13@gmail.com
Rams Junior	10990	engineering physicist	40	junirams@gmail.com
Britto francis	12333	instrument machinist	35	brittof@gmail.com
H P Waltmann	14325	mechanical technician	41	waltmannhp@gmail.com
J W Sims	16890	engineering physicist	55	jwsim43@gmail.com
A D Rushell	19876	instrument machinist	46	adrushell@gmail.com

Primary Key: p_sno

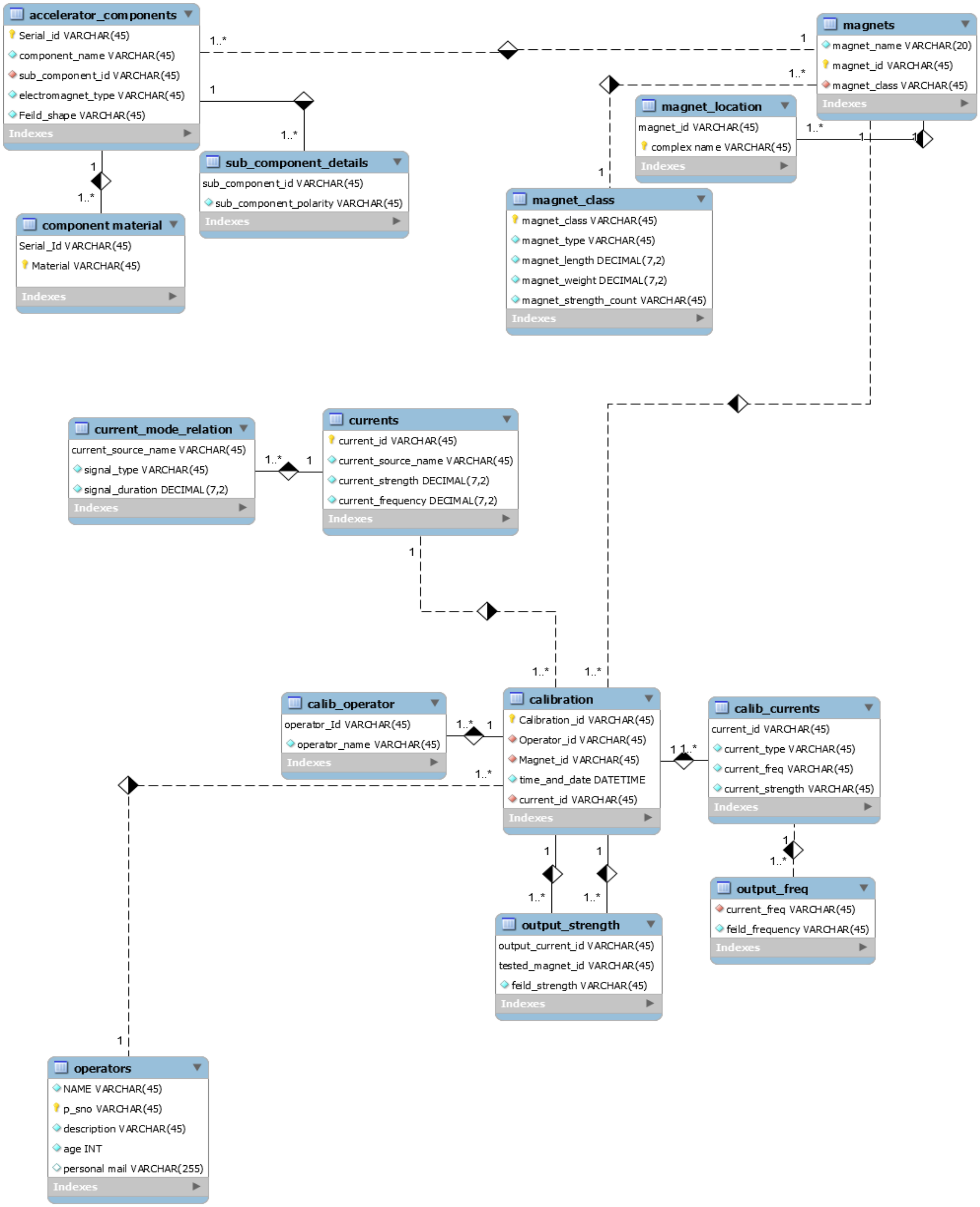
There are no non trivial dependencies, MVDs and join dependencies in this relation, moreover there are no multivalued and composite attributes, hence this table satisfies all 5 Normal forms.

Note:

After normalizing we get sub tables, foreign key constraints are created between these sub tables so that the information said by , the original table before normalization, is preserved

After normalization we can clearly observe that a lot of redundant data has been removed, hence database size have reduced.

ER diagram for my database after normalization:



SQL QUERIES:

1. Find the current source and magnet name used by the operators for each calibration performed.

   Query: select calibration_id, current_source_name, magnet_name from currents natural join calibration natural join magnets;

Table

calibration_id	current_source_name	magnet_name
C1234	F17 Kicker	alpha11
C2134	F17 Kicker	alpha23
C3223	E48 Kicker	staralpha
C4214	E48 Kicker	beta45
C4312	FO Lambertson	alpha34
C5432	E48 Kicker	gamma2
C5473	E48 Kicker	staralpha
C5678	FO Lambertson	beta23
C7123	E48 Kicker	gamma3
C8901	E48 Kicker	alpha11
C9087	F17 Kicker	beta33

2. Find the operators who has calibrated the same magnet more than once.

   Query: select operator_id,magnet_id from calibration group by magnet_id,operator_id having count(magnet_id)>1;

Table

operator_id	magnet_id
10101	M1210

He is the only operator who have calibrated the same magnet 'm1210' more than once in the recorded period.

3. Find the name and the class of magnets used for each accelerator component.

   Query: select component_name, magnet_name as "magnet_used", magnet_class from magnets, accelerator_components where magnet_id=sub_component_id;

Table

component_name	magnet_used	magnet_class
bending magnet	alpha11	A
focusing magnet	beta23	R
magnetic horn	staralpha	P
Undulator	gamma2	B
Solenoid magnet	gamma3	R
Booster kicker magnet	beta33	B
Storage ring magnet	staralpha	P

4. How many dipole magnets and how many quadrupole magnets are used in Fermilab.

Query: select magnet_type,count(magnet_type) as “no_of_magnets” from magnets natural join magnet_class group by magnet_type;

Table

magnet_type	no_of_magnets
Dipole	4
Quadrupole	6

5.Find out all the calibrations performed by engineering physicist, find out the operator name, magnet_name and the date of calibration

Query: select calibration_id, time_and_date , operator_name ,magnet_name from calibration natural join magnets ,operators where operator_id=p_sno and description="engineering physicist" ;

Table

calibration_id	time_and_date	operator_name	magnet_name
C4312	2015-06-12 12:03:34	Rams Junior	alpha34
C2134	2015-07-04 10:02:03	Harry bond	alpha23
C5432	2015-08-04 14:21:40	Harry bond	gamma2