



*2017 Draft - Prospect Analysis*  
**The New York Jets**

---

**BLAISE SPINELLI**

# Table of Contents

<b>Table of contents .....</b>	<b>2</b>
<b>Executive Summary.....</b>	<b>3</b>
<b>ER Diagram .....</b>	<b>4</b>
<b><u>Tables</u></b>	
<b>People .....</b>	<b>5</b>
<b>States .....</b>	<b>6</b>
<b>Coaches .....</b>	<b>7</b>
<b>Players.....</b>	<b>8</b>
<b>Schools.....</b>	<b>9</b>
<b>NFLCombineresults....</b>	<b>10</b>
<b>Backgroundcheck .....</b>	<b>11</b>
<b>Crimecommitted.....</b>	<b>12</b>
<b>MedicalHistory .....</b>	<b>13</b>
<b>Surgeries .....</b>	<b>14</b>
<b>Interviews.....</b>	<b>15</b>
<b><u>Views</u></b>	
<b>InterviewResults.....</b>	<b>16</b>
<b>PlayerSurgeries.....</b>	<b>17</b>
<b><u>Reports- Queries</u></b>	
<b>1st.....</b>	<b>19</b>
<b>Results.....</b>	<b>20</b>
<b>2<sup>nd</sup> .....</b>	<b>21</b>
<b>Results.....</b>	<b>22</b>
<b>3<sup>rd</sup> .....</b>	<b>23</b>
<b>Results.....</b>	<b>24</b>
<b><u>Stored Procedures</u></b>	
<b>Valid Combine Input() .....</b>	<b>25</b>
<b><u>Triggers</u></b>	
<b>Valid Combine Input .....</b>	<b>26</b>
<b><u>Security</u></b>	
<b>Executive Staff.....</b>	<b>27</b>
<b>Coaching Staff.....</b>	<b>28</b>
<b>Guests.....</b>	<b>29</b>
<b><u>Implementation Notes.....</u></b>	<b>30</b>
<b><u>Known Problems.....</u></b>	<b>30</b>
<b><u>Future Enhancements.....</u></b>	<b>30</b>



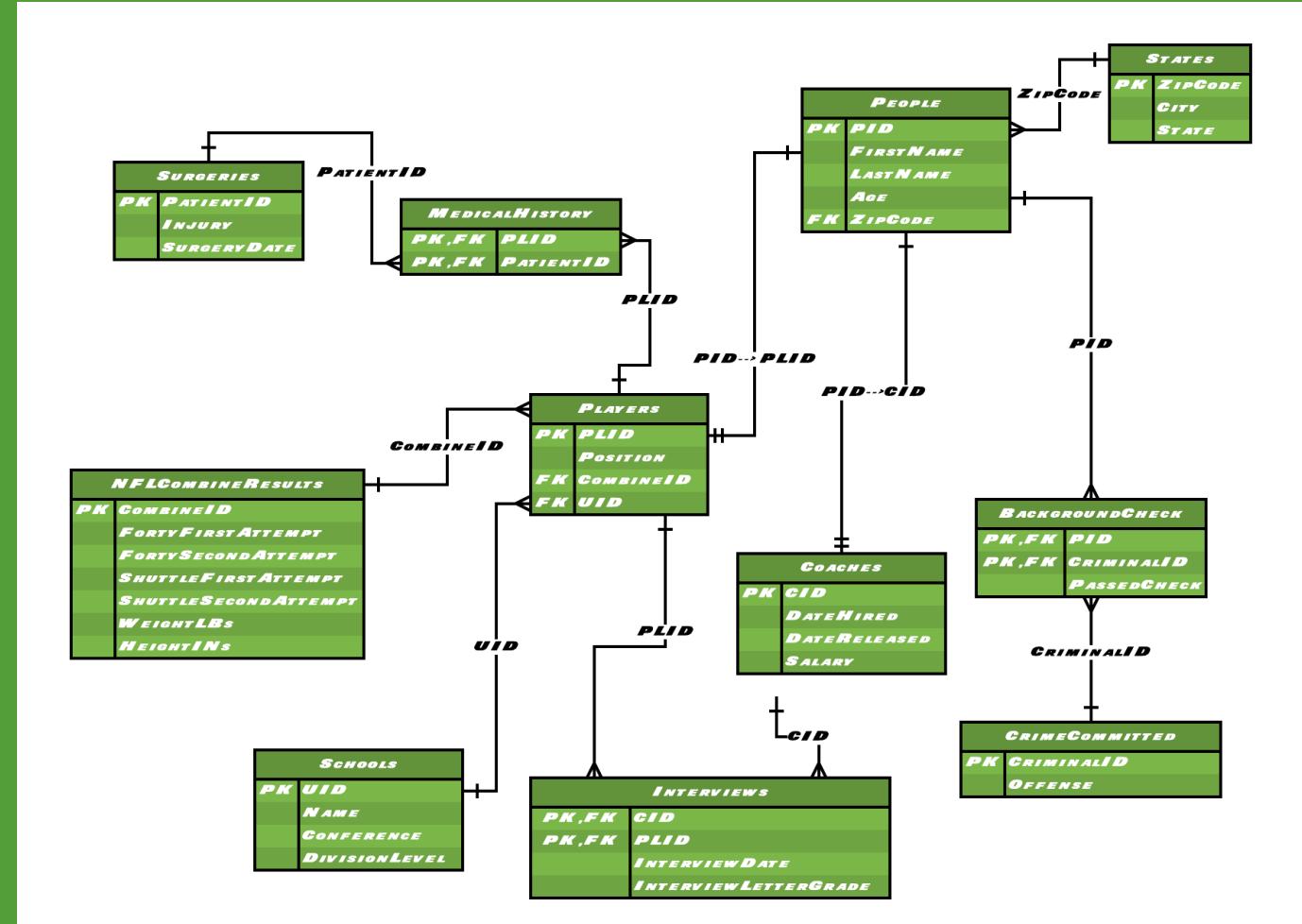
# Executive Summary

---

*As the 2017 NFL Draft approaches, The New York Jets hope to draft a promising group of prospects. It is crucial that the Jets bring in a talented group of men who can propel their organization to a Super Bowl appearance. Due to a lack of young talent brought in over recent years, a new database has been designed and implemented to ensure the Jets 2017 draft class is as talented and well rounded as possible. The ultimate objective is to establish a database that is not only fully functional, but also fully normalized in third normal form that can help the New York Jets properly analyze prospects. This database holds all the information required for each possible draft pick, including their alma mater, medical history, full background check, performances during the NFL Combine and even a formal interview. This information can be queried to narrow down the potential prospects worthy of being drafted. The data implemented into this database are primarily fictional.*

# Entity Relationship Diagram

BLAISE SPINELLI



# People Table

The people table contains all the people along with their common attributes  
The two subtypes of the people table are players and coaches



```
CREATE TABLE people (
    PID      char(5)      not null,
    FirstName  text        not null,
    LastName   text        not null,
    Age        integer     not null,
    ZipCode    integer     not null references states(ZipCode),
    primary key (PID)
);
```

## Functional Dependencies

$PID \rightarrow FirstName, LastName, Age, ZipCode$

**Sample Data:**

pid	character	firstname	lastname	age	zipcode
p1	Todd	Bowles		47	17032
p2	Mike	Caldwell		45	17032
p3	Brant	Boyer		45	17032
p4	Alan	Lahousseur		41	17032
p5	Kacy	Rodgers		42	17032
p6	Tim	Atkins		48	17032
p7	Jeremy	Bates		44	17032
p8	Ryan	Wilson		20	78701
p9	Dakota	Rogers		21	90001
p10	Jeremiah	Williams		20	60601
p11	Andrew	Reese		22	11798
p12	Tate	Armstrong		21	30301
p13	James	Cot		22	44101
p14	Austin	McDonald		20	33124
p15	Rara	Cucumila		21	15201
p16	Darnell	King		20	37201
p17	Titus	Booker		21	97201
p18	Jamal	Freeman		20	11798
p19	James	Tyler		22	30301
p20	Scott	Thomson		21	60601
p21	Martin	Harris		20	11798
p22	Bryant	Cobb		21	30301
p23	Trey	Sengs		20	44101
p24	Drake	Ova		22	33124
p25	Owen	Charles		21	15201
p26	Mike	Coyle		22	37201
p27	Roll	Mora		20	97201
p28	Justin	Brackler		21	78701
p29	Derek	Merck		20	90001
p30	Andre	Gomez		20	60601
p31	Nick	Dip		21	11798
p32	YaYa	Sewid		22	30301
p33	Brandon	Bond		20	44101
p34	Brady	Bunch		21	33124
p35	Thomas	Gallo		20	15201
p36	LaQuan	Noil		21	37201
p37	Johnny	Oats		22	97201



# States Table

Lists all Cities and States associated with the corresponding Zip Code

```
CREATE TABLE states (
    ZipCode      integer      not null,
    City         text         not null,
    State        text         not null,
    primary key (ZipCode)
);
```

Functional Dependencies  
**ZipCode → City, State**

**Sample  
Data:**

	zipcode integer	city text	state text
	17032	Hoboken	New Jers...
	90001	Los Ange...	California
	60601	Chicago	Illinois
	11798	Wheatley...	New York
	30301	Atlanta	Georgia
	44101	Cleveland	Ohio
	78701	Austin	Texas
	33124	Miami	Florida
	15201	Pittsburgh	Pennsylv...
	37201	Nashville	Tennessee
	97201	Portland	Oregon



# Coaches Table

The Coaches Table provides the date the coach was hired, their salary, and when or if they were released

```
CREATE TABLE coaches (
    CID          char(5)      not null unique references people(PID),
    DateHired    date         not null,
    DateReleased date,
    SalaryUSD    integer       not null,
    primary key (CID)
);
```

## Functional Dependencies

$CID \rightarrow DateHired, DateReleased, SalaryUSD$

**Sample  
Data:**

	cid character	datehired date	daterelea... date	salaryusd integer
	p1	2014-03-...		3000000
	p2	2014-03-...		2500000
	p3	2014-03-...		1700000
	p4	2014-03-...		8500000
	p5	2014-04-...		1000000
	p6	2014-04-...		500000
	p7	2014-04-...		600000



# Players Table

Lists all players and their attributes that coaches should be aware of

```
CREATE TABLE players (
    PLID      char(5)      not null unique references people(PID),
    position   text         not null,
    CombineID  char(5)      not null unique references nflcombineresults(CombineID),
    UID        char(5)      not null unique references schools(UID),
    primary key (PLID),
    foreign key (CombineID) references nflcombineresults(CombineID),
    foreign key (UID) references schools(UID)
);
```

## Functional Dependencies

$PLID \rightarrow position, CombineID, UID$

**Sample  
Data:**

	plid character	position text	combineid character	uid character
	p8	RB	p8	p8
	p9	QB	p9	p9
	p10	WR	p10	p10
	p11	WR	p11	p11
	p12	OL	p12	p12
	p13	OL	p13	p13
	p14	OL	p14	p14
	p15	DB	p15	p15
	p16	DB	p16	p16
	p17	DB	p17	p17
	p18	LB	p18	p18
	p19	LB	p19	p19
	p20	QB	p20	p20
	p21	RB	p21	p21
	p22	DB	p22	p22
	p23	WR	p23	p23
	p24	DL	p24	p24
	p25	DL	p25	p25
	p26	RB	p26	p26
	p27	QB	p27	p27
	p28	LB	p28	p28
	p29	WR	p29	p29
	p30	DL	p30	p30
	p31	QB	p31	p31
	p32	OL	p32	p32
	p33	RB	p33	p33
	p34	LB	p34	p34
	p35	DL	p35	p35
	p36	QB	p36	p36
	p37	QB	p37	p37



# Schools Table

Lists all schools players attended

```
CREATE TABLE schools (
    UID          char(5)      not null unique,
    name         text         not null unique,
    conference   text,
    divisionlevel char(5)      not null,
primary key(UID)
);
```

## Functional Dependencies

$UID \rightarrow name, conference, divisionlevel$

**Sample  
Data:**

	uid character	name text	conference text	divisionle... character
	p8	LSU	SEC	1
	p9	FSU	ACC	1
	p10	TCU	Big 12	1
	p11	Oregon	Pac 12	1
	p12	Auburn	SEC	3
	p13	Vanderbilt	SEC	1
	p14	Michigan	Big 10	1
	p15	Penn State	Big 10	1
	p16	Arizona St...	Pac 12	1
	p17	Miami	ACC	1
	p18	Clemson	ACC	1
	p19	Iowa	Big 10	1
	p20	Texas	Big 12	1
	p21	Oklahoma	Big 12	1
	p22	Georgia	SEC	1
	p23	Alabama	SEC	1
	p24	Nebraska	Big 10	1
	p25	Boston Co...	ACC	1
	p26	Ohio State	Big 10	1
	p27	USC	Pac 12	1
	p28	Baylor	Big 12	1
	p29	Utah	Pac 12	1
	p30	Pitt	ACC	1
	p31	Kansas	Big 12	1
	p32	Colorado	Pac 12	1
	p33	Duke	ACC	1
	p34	Kentucky	SEC	1
	p35	Wisconsin	Big 10	1
	p36	UCLA	Pac 12	1
	p37	West Virg...	Big 12	1

# NFLCombineResults Table



Lists all measurables & stats recorded by players at the NFL Combine for the forty yard dash and shuttle run

```
CREATE TABLE nflcombineresults (
    CombineID      char(5)      not null,
    fortyfirstattempt  char(5),
    fortysecondattempt  char(5),
    shuttlefirstattempt  char(5),
    shuttlesecondattempt  char(5),
    weightLBs        integer     not null,
    heightIns        integer     not null,
    primary key(CombineID)
);
```

Sample  
Data:

	combinedcharacter	fortyfirstcharacter	fortyseconcharacter	shuttlefirstcharacter	shuttleseccharacter	weightlbsinteger	heightinsinteger
	p8	4.59	4.58	4.00	4.01	212	73
	p9	4.82	4.90	4.36	4.31	226	75
	p10	4.33	4.38	4.01	3.91	197	71
	p11	4.66	4.62	4.22	4.25	246	74
	p12	4.92	5.01	4.65	4.60	308	73
	p13	4.99	5.13	4.55	4.64	313	77
	p14	5.29	5.23	4.71	4.69	311	78
	p15	4.22	4.20	3.90	3.99	208	71
	p16	4.49	4.45	4.10	4.09	213	73
	p17	4.25	4.22	4.02	4.00	215	75
	p18	4.50	4.52	4.13	4.17	236	74
	p19	4.72	4.72	4.33	4.27	242	73
	p20	4.52	4.51	4.13	4.19	211	73
	p21	4.40	4.37	4.03	4.19	202	70
	p22	4.56	4.57	4.23	4.19	213	73
	p23	4.26	4.27	4.00	4.00	193	71
	p24	4.79	4.72	4.40	4.45	289	74
	p25	4.52	4.53	4.20	4.25	282	73
	p26	4.22	4.25	3.82	3.95	202	72
	p27	4.52	4.55	4.02	4.05	219	74
	p28	4.58	4.50	4.18	4.15	227	74
	p29	4.58	4.63	4.28	4.30	211	73
	p30	4.48	4.49	4.30	4.27	271	72
	p31	4.78	4.79	4.49	4.57	230	76
	p32	4.98	5.09	4.69	4.67	320	77
	p33	4.28	4.24	3.89	3.77	208	71
	p34	4.58	4.54	4.19	4.20	225	71
	p35	4.78	4.84	4.49	4.50	305	73
	p36	4.98	4.92	4.59	4.50	227	76
	p37	4.48	4.42	4.09	4.10	227	75

## Functional Dependencies

$\text{CombineID} \rightarrow \text{fortyfirstattempt}$ ,  
 $\text{fortysecondattempt}$ ,  $\text{shuttlefirstattempt}$ ,  
 $\text{shuttlesecondattempt}$ ,  $\text{weightLBs}$ ,  $\text{heightIns}$



# Backgroundcheck Table

States whether or not the player has passed a background check

```
CREATE TABLE backgroundcheck (
    PID          char(5)      not null references people(PID),
    passedcheck  boolean      not null,
    criminalID   char(5)      not null references crimecommitted(criminalID),
primary key(PID,criminalID)
);
```

Functional Dependencies  
 $PID, CriminalID \rightarrow passedcheck$

Sample  
Data:

pid character	passedch.. boolean	criminalid character
p8	0	cr09
p9	0	cr10
p10	0	cr11
p11		cr01
p12	0	cr12
p13	0	cr13
p14		cr02
p15		cr03
p16	0	cr14
p17	0	cr15
p18		cr04
p19	0	cr16
p20		cr17
p21	0	cr18
p22		cr19
p23		cr20
p24		cr05
p25	0	cr21
p26		cr22
p27		cr23
p28	0	cr24
p29		cr06
p30	0	cr25
p31	0	cr26
p32		cr07
p33	0	cr27
p34	0	cr28
p35		cr08
p36	0	cr29
p37	0	cr30



# Crimecommitted Table

Table indicating the crime that was committed by the player

```
CREATE TABLE crimecommitted (
    criminalID      char(5)      not null,
    offense          text         not null,
    primary key(criminalID)
);
```

Functional Dependencies  
 $\text{CriminalID} \rightarrow \text{offense}$

Sample  
Data:

	criminalid	offense
	character	text
	cr01	sexual ass...
	cr02	theft
	cr03	dui
	cr04	drugs
	cr05	theft
	cr06	theft
	cr07	sexual ass...
	cr08	dui



# MedicalHistory Table

Lists the players *PatientID* if they underwent surgery

```
CREATE TABLE medicalhistory (
    PLID          char(5)
    patientID     char(5)
primary key(PLID,patientID)
);
not null references players(PLID),
references surgeries(patientID),
```

**Sample  
Data:**

plid character	patientid character
p8	pat1
p9	pat2
p10	pat3
p11	pat4
p12	pat5
p13	pat23
p14	pat6
p15	pat24
p16	pat7
p17	pat25
p18	pat26
p19	pat8
p20	pat27
p21	pat9
p22	pat28
p23	pat10
p24	pat11
p25	pat12
p26	pat13
p27	pat29
p28	pat30
p29	pat14
p30	pat15
p31	pat16
p32	pat17
p33	pat18
p34	pat19
p35	pat20
p36	pat21
p37	pat22

**Functional Dependencies**  
 $PLID, PatientID \rightarrow N/A$

# Surgeries Table

Lists the Player's PatientID and the injury that required surgery, dates also included



```
CREATE TABLE surgeries (
    patientID          char(5),
    injury              text,
    surgerydate         date,
    primary key(patientID)
);
```

Sample  
Data:

patientid character	injury text	surgeryda... date
pat1	Torn ACL	2017-01-03
pat2	Torn MCL	2016-12-13
pat3	Torn ACL	2014-10-23
pat4	Torn PCL	2015-03-30
pat5	Torn ACL	2016-11-03
pat6	Torn labru...	2017-02-05
pat7	Torn labru...	2014-04-03
pat8	Broken Leg	2016-07-03
pat9	Torn ACL	2015-01-03
pat10	Broken no...	2014-12-03
pat11	Broken leg	2014-06-03
pat12	Torn MCL	2014-07-23
pat13	Torn MCL	2016-08-13
pat14	Torn ACL	2013-09-03
pat15	Broken no...	2015-02-23
pat16	Broken arm	2016-04-03
pat17	Torn ACL	2015-01-13
pat18	Torn labru...	2014-10-13
pat19	Torn labru...	2015-11-13
pat20	Torn ACL	2016-12-03
pat21	Broken leg	2016-04-23
pat22	Torn MCL	2016-05-23
pat23	Torn PCL	2016-05-13
pat24	Broken arm	2014-01-03
pat25	Broken no...	2015-02-13
pat26	Torn PCL	2015-11-03
pat27	Torn MCL	2013-12-23
pat28	Torn ACL	2016-08-23
pat29	Broken no...	2015-09-13
pat30	Broken arm	2016-09-03

## Functional Dependencies

$\text{patientID} \rightarrow \text{injury}, \text{surgerydate}$



# Interviews Table

*States the grade and the date of the interview conducted by coaches*

```
CREATE TABLE interviews (
    CID          char(5)      not null references coaches(CID),
    PLID         char(5)      not null references players(PLID),
    interviewdate date,
    interviewlettergrade char(2),
    primary key(CID,PLID)
);
```

**Sample  
Data:**

	cid character	plid character	Interview... date	Interview... character
	p4	p8	2017-03-10	B
	p3	p9	2017-03-10	B
	p2	p10	2017-03-10	C
	p2	p11	2017-03-10	A
	p3	p12	2017-03-10	A
	p1	p13	2017-03-10	C
	p3	p14	2017-03-10	A
	p1	p15	2017-03-10	A
	p1	p16	2017-03-10	B
	p3	p17	2017-03-10	B
	p7	p18	2017-03-10	B
	p2	p19	2017-03-10	C
	p3	p20	2017-03-10	A
	p1	p21	2017-03-10	C
	p5	p22	2017-03-10	F
	p3	p23	2017-03-10	A
	p3	p24	2017-03-10	A
	p3	p25	2017-03-10	C
	p3	p26	2017-03-10	F
	p7	p27	2017-03-10	B
	p1	p28	2017-03-10	B
	p4	p29	2017-03-10	B
	p6	p30	2017-03-10	F
	p3	p31	2017-03-10	B
	p2	p32	2017-03-10	A
	p1	p33	2017-03-10	B
	p5	p34	2017-03-10	C
	p7	p35	2017-03-10	B
	p2	p36	2017-03-10	B
	p3	p37	2017-03-10	A

## Functional Dependencies

$CID,PLID \rightarrow interviewdate,interviewlettergrade$

# Views



**InterviewResults** - Allows coaches to see how each player performed during the formal interview

```
DROP VIEW IF EXISTS InterviewResults;
CREATE VIEW InterviewResults AS
SELECT people.firstname, people.lastname, interviews.interviewlettergrade
FROM people
INNER JOIN players ON players.plid = people.pid
INNER JOIN interviews ON interviews.plid = players.plid
ORDER BY interviews.interviewlettergrade DESC
;
```

**InterviewResults can be Queried to narrow down which players the coaching staff thought were the best formal interviewees**

```
SELECT *
FROM InterviewResults
WHERE interviewlettergrade = 'A'
;|
```

**Sample Output:**

	firstname text	lastname text	intervie... character
	Andrew	Reese	A
	Tate	Armstrong	A
	Austin	McDonald	A
	Rara	Cucumba	A
	Scott	Thomson	A
	Trey	Sengs	A
	Drake	Ovo	A
	YaYa	Sewld	A
	Johnny	Oats	A

# Views



## PlayerSurgeries – Allows easy access for coaches to see what players had surgery

```
DROP VIEW IF EXISTS PlayerSurgeries;
CREATE VIEW PlayerSurgeries as
select people.firstname, people.lastname, surgeries.surgerydate, surgeries.injury
from people
inner join players on players.plid = people.pid
inner join medicalhistory on medicalhistory.plid = players.plid
inner join surgeries on surgeries.patientid = medicalhistory.patientid
;
```

If a coach wants to know the specific injury which required surgery to be performed, they can query this view for that information. An example is to the right

```
select *
from ACLSurgeries
where injury = 'Torn ACL'
;
```

### Sample Output:

	firstname text	lastname text	surgeryd... date	injury text
	Ryan	Wilson	2017-01...	Torn ACL
	Jeremiah	Williams	2014-10...	Torn ACL
	Tate	Armstrong	2016-11...	Torn ACL
	Martin	Harris	2015-01...	Torn ACL
	Derek	Merck	2013-09...	Torn ACL

# Views



**Players\_with\_criminal\_record** – Indicates all players who have pleaded guilty to committing a crime

```
DROP VIEW IF EXISTS Players_with_criminal_record;
CREATE VIEW Players_with_criminal_record as
select crimecommitted.offense, backgroundcheck.pid, backgroundcheck.criminalID, people.firstname, people.lastname,
from crimecommitted
inner join backgroundcheck on backgroundcheck.criminalID = crimecommitted.criminalID
inner join people on people.pid = backgroundcheck.pid
; 
```

**Queries can be used to discover the type of the offense that a prospect has made, the query to the right displays those who have been convicted of theft**

```
select *
from Players_with_criminal_record
where passedcheck = false and
offense = 'theft'
order by firstname
;| 
```

**Sample Output:**

	offense text	pid character	criminalid character	firstname text	lastname text	passedc... boolean
	theft	p14	cr02	Austin	McDonald	
	theft	p29	cr06	Derek	Merck	
	theft	p24	cr05	Drake	Ovo	

# Reports – Interesting Queries

These are queries that demonstrate the analytical potential of databases. These are basic examples, but nonetheless examples of the kinds of information that one can extrapolate from data. The New York Jets 2017 Super Bowl hopes rely heavily on the output of these queries



1. The Jets believe that bringing in taller players may increase their chances of success this season, so a Query was created that returns all players who are at least 6'2"

```
select nflcombineresults.heightIns, people.firstname, people.lastname, players.position, schools.name, schools.conference  
from players  
inner join schools on schools.uid = players.plid  
inner join people on people.pid = players.plid  
inner join nflcombineresults on nflcombineresults.combineID = players.plid  
where nflcombineresults.heightIns > '73'  
order by nflcombineresults.heightIns DESC  
;
```

Results of this Query are located on the following page



# Reports – Interesting Queries

Results from 1<sup>st</sup> Query:

	heightin Integer	firstname text	lastname text	position text	name text	conferen... text
	78	Austin	McDonald	OL	Michigan	Big 10
	77	YaYa	Sewid	OL	Colorado	Pac 12
	77	James	Cot	OL	Vanderbilt	SEC
	76	Nick	Dip	QB	Kansas	Big 12
	76	LaQuan	Noll	QB	UCLA	Pac 12
	75	Dakota	Rogers	QB	FSU	ACC
	75	Titus	Booker	DB	Miami	ACC
	75	Johnny	Oats	QB	West Vir...	Big 12
	74	Justin	Brackler	LB	Baylor	Big 12
	74	Drake	Ovo	DL	Nebraska	Big 10
	74	Jamal	Freeman	LB	Clemson	ACC
	74	Andrew	Reese	WR	Oregon	Pac 12
	74	Rob	Mora	QB	USC	Pac 12



# Reports – Interesting Queries

**2. In the 2016 season, The Jets had one glaring flaw—speed. With a plethora of young speedsters in this years draft, the Jets would like to analyze the speed of the incoming talent, starting with who ran the fastest 40 yard dash time at the NFL combine. The following Query returns which players ran under a 4.5 second 40 yard dash time at the combine.**

```
select nflcombineresults.fortyfirstattempt, people.firstname, people.lastname, players.position, schools.name, schools.conference  
from players  
inner join nflcombineresults on nflcombineresults.combineid = players.plid  
inner join people on people.pid = players.plid  
inner join schools on schools.uid = players.plid  
where nflcombineresults.fortyfirstattempt < '4.50'  
order by nflcombineresults.fortyfirstattempt ASC  
;
```

*Results of this Query are located on the following page*



# Reports – Interesting Queries

Results from 2<sup>nd</sup> Query:

	fortyfirst... character	firstname text	lastname text	position text	name text	conferen... text
	4.22	Rara	Cucumba	DB	Penn State	Big 10
	4.22	Mike	Coyle	RB	Ohio State	Big 10
	4.25	Titus	Booker	DB	Miami	ACC
	4.26	Trey	Sengs	WR	Alabama	SEC
	4.28	Brandon	Bond	RB	Duke	ACC
	4.33	Jeremiah	Williams	WR	TCU	Big 12
	4.40	Martin	Harris	RB	Oklahoma	Big 12
	4.48	Andre	Gomez	DL	Pitt	ACC
	4.48	Johnny	Oats	QB	West Vir...	Big 12
	4.49	Darnell	King	DB	Arizona S...	Pac 12



# Reports – Interesting Queries

**3. The Jets believe that some of their current coaches may be getting compensated a little too heavily. In hopes to clear up cap space, the Jets announce that pay cuts must be made. This Query will allow the Jets organization to see each coach's salary so the proper reductions are made**

```
select people.firstname, people.lastname, people.age, states.state, states.city, coaches.salaryusd  
from people  
inner join states on states.zipcode = people.zipcode  
inner join coaches on people.pid = coaches.cid  
order by coaches.salaryusd DESC  
;
```

*Results of this Query are located on the following page*



# Reports – Interesting Queries

Results from 3<sup>rd</sup> Query:

	firstname text	lastname text	age integer	state text	city text	salaryusd integer
	Alan	Labouseur	41	New Jers...	Hoboken	8500000
	Todd	Bowles	47	New Jers...	Hoboken	3000000
	Mike	Caldwell	45	New Jers...	Hoboken	2500000
	Brant	Boyer	45	New Jers...	Hoboken	1700000
	Kacy	Rodgers	42	New Jers...	Hoboken	1000000
	Jeremy	Bates	44	New Jers...	Hoboken	600000
	Tim	Atkins	48	New Jers...	Hoboken	500000

# Stored Procedures



**Valid Combine Input** – This stored procedure will check to make sure that a player has participated in at least one of the events at the combine before they are entered into the database. It will also ensure that their height and weight was recorded as well

```
CREATE OR REPLACE FUNCTION Valid_combine_input()
RETURNS TRIGGER AS
$$ BEGIN
IF NEW.fortyfirstattempt IS NULL and NEW.shuttlefirstattempt IS NULL THEN
RAISE EXCEPTION 'player must have at least one recorded event';
END IF;
IF NEW.weightLBs IS NULL and NEW.heightIns IS NULL THEN
RAISE EXCEPTION 'height and weight must be recorded at the combine';
END IF;
RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

*The sample output for this is with the Valid\_combine\_input Trigger on following page*

# Triggers



**Valid Combine Input** – This trigger will execute the `Valid_combine_input()` stored procedure anytime an insert or update is made to the `Nflcombineresults` table.

**Trigger:**

```
CREATE TRIGGER Valid_combine_input
BEFORE INSERT OR UPDATE ON nflcombineresults
FOR EACH ROW
EXECUTE PROCEDURE Valid_combine_input();
```

**Sample Input:**

```
INSERT INTO nflcombineresults
values
('p35',null,'4.50',null,'4.4','74','300');
```

**Sample output:**

```
ERROR: player must have at least one recorded event
CONTEXT: PL/pgSQL function valid_combine_input() line 3 at RAISE
***** Error *****
```

```
ERROR: player must have at least one recorded event
SQL state: P0001
Context: PL/pgSQL function valid_combine_input() line 3 at RAISE
```



# Security

## Executive Staff

**The New York Jets Executive staff will be granted access to the entire 2017 database so all information on incoming prospects can be evaluated.**

```
CREATE ROLE ExecStaff;  
GRANT ALL ON ALL TABLES  
IN SCHEMA PUBLIC  
TO ExecStaff;
```



# Security

## Coaching Staff

***The New York Jets Coaching staff will be granted access to the entire 2017 database so all information on incoming prospects can be evaluated.***

```
CREATE ROLE CoachingStaff;  
GRANT ALL ON ALL TABLES  
IN SCHEMA PUBLIC  
TO CoachingStaff;
```



# Security

## Guests

***Guests will be granted very minimal access to ensure that specific details about the players are not leaked out to the media.***

```
CREATE ROLE Guests;
GRANT SELECT ON players,
               coaches,
               medicalhistory,
               surgeries,
               schools,
               nflcombineresults
TO Guests;
```

# **Implementation Notes, Known Problems, and Future Enhancements**



***It is important to note a few characteristics of this particular database. Obviously, a real scale database of the prospects who are eligible for the draft would contain roughly ten times the amount of data. There are dozens of characteristics and factors that play into the real life analysis of a college prospect. One of the most telling signs of a college players' chance of success in the NFL are their stats that they recorded throughout their collegiate career, which was not accounted for.***

***A few problems do exists within this database that should be addressed. More views should exist that allow coaches to readily access information about how the player tested at the NFL combine. The NFL combine results table also lacks events that play a significant role in the players draft stock.***

***A few future enhancements include the implementation of multiple check constraints. The first being a check for a players current eligibility in the NCAA. If a player is deemed ineligible by the NCAA, an organization may not be willing to draft them.***