STORED PROCEDURES

ADD CANDIDATE ROW

CREATE DEFINER=`root`@`localhost` PROCEDURE `addcandidatesrow`(IN candidateid INT(11),

IN candname VARCHAR(45),

IN candaddress VARCHAR(45),

IN candtelephone VARCHAR(45),

IN userskill_id1 VARCHAR(45),

IN userskill_id2 VARCHAR(45),

IN userskill_id3 VARCHAR(45))

BEGIN

INSERT INTO candidates(candidateid, candname, candaddress, candtelephone)
VALUES(candidateid, candname, candaddress, candtelephone);

INSERT INTO haveskills(fk_candidateid, fk_skillid) VALUES(candidateid, userskill_id1), (candidateid, userskill_id2), (candidateid, userskill_id3); END

candidateid	15	[IN]	INT(11)
candname	barry	[IN]	VARCHAR(45)
candaddress	olsen	[IN]	VARCHAR(45)
candtelephone	4356432	[IN]	VARCHAR(45)
userskill_id1	admin	[IN]	VARCHAR(45)
userskill_id2	cod	[IN]	VARCHAR(45)
userskill_id3	eng	[IN]	VARCHAR(45)

CANDIDATE TABLE AFTER QUERY

HAVESKILLS SKILLS TABLE AFTER QUERY

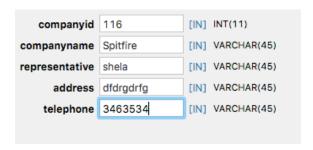
▶	1	sarah	fitzpatrick	13254657890768
	2	paul	lyons	13245487645674
	3	ryan	morris	54677432457654
	4	bill	durden	087856434567657677
	5	jessica	blanc	3645353452423233
	6	rachel	jolie	453423446565645567
	7	cameron	pitt	56432656346246
	8	kevin	murphy	97675674676574
	9	philip	fitzgibbons	24325436453234
	10	gary	raythorn	754646435343645
	15	barry	olsen	4356432
	NULL	NULL	NULL	NULL

9	admin
9	eng
9	sal
10	admin
10	eng
10	sal
15	admin
15	cod
15	eng
NULL	NULL

ADD COMPANY ROW

CREATE DEFINER=`root`@`localhost` PROCEDURE `addcompanyrow`(IN companyid INT(11), IN companyname VARCHAR(45), IN representative VARCHAR(45), IN address VARCHAR(45), IN telephone VARCHAR(45))
BEGIN

INSERT INTO companies(companyid, companyname, representative, address, telephone)
VALUES(companyid, companyname, representative, address, telephone);
END



▶	100	apple	carry	fake street	7536474
	101	google	billbo	Ballymun	9647653
	102	microsoft	gandalf	Detroit	7246457
	103	dell	ron	Los Angeles	8457453
	104	aspiron	penelope	San Francisco	1374564
	105	acer	justin	Germany	0785645
	106	lenovo	steven	London	7574643
	107	hp	paula	Essex	9568754
	108	dropbox	jenn	Dublin	8456453
	109	ebay	venessa	Galway	8346345
	110	amazon	helena	Berlin	7346342
	115	Smiths	barry	dggrsrdf	4574564
	116	Spitfire	shela	dfdrgdrfg	3463534

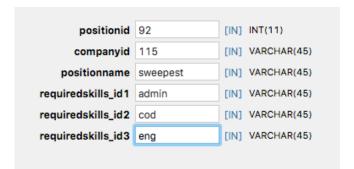
ADD POSITION ROW

CREATE DEFINER=`root`@`localhost` PROCEDURE `addpositionrow`(IN positionid INT(11), IN companyid VARCHAR(45), IN positionname VARCHAR(45), IN requiredskills_id1 VARCHAR(45), IN requiredskills_id2 VARCHAR(45), IN requiredskills_id3 VARCHAR(45), IN requiredskills_id3 VARCHAR(45))

BEGIN

INSERT INTO positions(positionid, fkcompanynid, positionname)
VALUES(positionid, companyid, positionname);

INSERT INTO requiredskills(fkpositionid, fkskillid)
VALUES(positionid, requiredskills_id1),
(positionid, requiredskills_id2),
(positionid, requiredskills_id3);
END



POSITION TABLE AFTER QUERY

REQUIRED SKILLS TABLE AFTER QUERY

80	100	programmer
81	101	manager
82	102	sales
83	103	Programmer
84	104	Manager
85	105	Sales
86	106	Programmer
87	107	Manager
88	108	Sales
89	109	Programming
90	110	Manager
91	115	ceo
92	115	sweepest
	81 82 83 84 85 86 87 88 89 90	81 101 82 102 83 103 84 104 85 105 86 106 87 107 88 108 89 109 90 110 91 115

63	eng
83	man
91	admin
91	cod
91	eng
92	admin
92	cod
92	eng
NULL	NULL

ADD SKILLS ROW

CREATE DEFINER=`root`@`localhost` PROCEDURE `addskillsrow`(IN skillid VARCHAR(45), IN skillname VARCHAR(45)) BEGIN

INSERT INTO skills(skillid, skillname) VALUES(skillid, skillname); END





ADD HIRED ROW

CREATE DEFINER=`root`@`localhost` PROCEDURE `addhiredrow`(IN candidateid VARCHAR(45), IN positionid VARCHAR(45)) BEGIN

INSERT INTO hiredcandidates(fk3_candidateid, fk3_positionid)
VALUES(candidateid, positionid);
END



⊳ 3	80	
4	80	
9	82	
1	83	
2	84	
5	85	
6	86	
7	87	
8	89	
10	90	
15	92	
NULL	NULL	

CANDIDATE WHO HAS SKILLS FROM POSITION ID

CREATE DEFINER=`root`@`localhost` PROCEDURE `cand_have_req_skill_from_posid`(IN positionid INT(11))

BEGIN

SELECT DISTINCT *

FROM candidates

WHERE candidateid IN (SELECT fk_candidateid

FROM haveskills

WHERE fk_skillid IN(SELECT fkskillid

FROM requiredskills

WHERE fkpositionid=positionid))

ORDER BY candidateid;

END



	candidataid	candname	candaddrass	candtelephone
	carididateid			
Þ	1	sarah	fitzpatrick	13254657890768
	2	paul	lyons	13245487645674
	3	ryan	morris	54677432457654
	4	bill	durden	087856434567657677
	6	rachel	jolie	453423446565645567
	7	cameron	pitt	56432656346246
	15	barry	olsen	4356432

CANDIDATE WITH GIVEN ID

CREATE DEFINER=`root`@`localhost` PROCEDURE `candidate_with_givenid`(IN inputcandidateid VARCHAR(45))

BEGIN

SELECT * FROM candidates WHERE candidateid=inputcandidateid; END



COMPANY WITH GIVEN NAME

CREATE DEFINER=`root`@`localhost` PROCEDURE `candidate_with_given_name`(IN candidatename VARCHAR(45))

BEGIN

SELECT * FROM candidates WHERE candname=candidatename; END



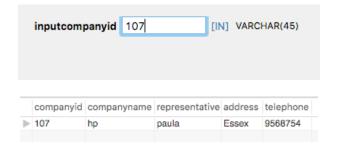
CANDIDATE WITH 2+ INTERVIEWS

CREATE DEFINER=`root`@`localhost` PROCEDURE `cand__who_have_2+_interviews`()
BEGIN
SELECT *
FROM candidates
WHERE candidateid IN(SELECT fk2_candidateid
FROM interviews
GROUP BY fk2_candidateid
HAVING COUNT(*) >= 2);
END

	candidateid	candname	candaddress	candtelephone
Þ	1	sarah	fitzpatrick	13254657890768
	2	paul	lyons	13245487645674
	3	ryan	morris	54677432457654
	4	bill	durden	087856434567657677
	5	jessica	blanc	3645353452423233
	6	rachel	jolie	453423446565645567
	7	cameron	pitt	56432656346246
	8	kevin	murphy	97675674676574
	9	philip	fitzgibbons	24325436453234
	10	gary	raythorn	754646435343645

COMPANY WITH GIVEN ID

CREATE DEFINER=`root`@`localhost` PROCEDURE `company_with_givenid`(IN inputcompanyid VARCHAR(45))
BEGIN
SELECT * FROM companies WHERE companyid=inputcompanyid;
END



COMPANY WITH GIVEN NAME

CREATE DEFINER=`root`@`localhost` PROCEDURE `company_with_givenname`(IN inputcompanyname VARCHAR(45))

BEGIN

SELECT * FROM companies WHERE companyname=inputcompanyname; END

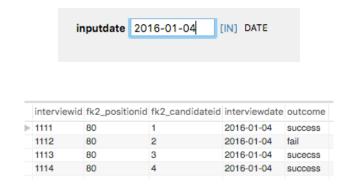


INTERVIEW ON GIVEN DATE

CREATE DEFINER=`root`@`localhost` PROCEDURE `interviews_on_specificdate`(IN inputdate DATE)

BEGIN

SELECT * FROM interviews WHERE interviewdate=inputdate; END



ORDER POSITIONS

CREATE DEFINER=`root`@`localhost` PROCEDURE `order_positions`()
BEGIN
SELECT * FROM positions ORDER BY fkcompanynid;
END

	positionid	fkcompanynid	positionname
>	80	100	programmer
	81	101	manager
	82	102	sales
	83	103	Programmer
	84	104	Manager
	85	105	Sales
	86	106	Programmer
	87	107	Manager
	88	108	Sales
	89	109	Programming
	90	110	Manager
	91	115	ceo
	92	115	sweepest

FIND POSITION FROM SKILL

CREATE DEFINER=`root`@`localhost` PROCEDURE `position_from_skillid`(IN skillid VARCHAR(45))

BEGIN

SELECT*

FROM positions

WHERE positionid IN(SELECT fkpositionid

FROM requiredskills

WHERE fkskillid=skillid);

END



POSITIONS THAT REQUIRE ADMIN

CREATE DEFINER=`root`@`localhost` PROCEDURE `positions_require_admin`()
BEGIN
SELECT DISTINCT COUNT(fkpositionid)
FROM requiredskills
WHERE fkskillid='admin';
END



POSITIONS FROM GIVEN ID

 $\label{local-position} CREATE\ DEFINER=`root`@`localhost`\ PROCEDURE\ `position_with_givenid`(IN\ enterpositionid\ INT(11))$

BEGIN

SELECT*

FROM positions

WHERE positionid=enterpositionid;

END





CANDIDATES

	candidateid	candname	candaddress	candtelephone
>	1	sarah	fitzpatrick	13254657890768
	2	paul	lyons	13245487645674
	3	ryan	morris	54677432457654
	4	bill	durden	087856434567657677
	5	jessica	blanc	3645353452423233
	6	rachel	jolie	453423446565645567
	7	cameron	pitt	56432656346246
	8	kevin	murphy	97675674676574
	9	philip	fitzgibbons	24325436453234
	10	gary	raythorn	754646435343645
	15	barry	olsen	4356432
	NULL	NULL	NULL	NULL

HAVESKILLS

	TAY LOIG	LLLO
	fk_candidateid	fk_skillid
▶	1	cod
	1	eng
	1	lead
	2	hyp
	2	know
	2	speak
	3	hyp
	3	know
	3	speak
	4	cod
	4	know
	4	rep
	5	lead
	5	pres
	5	speak
	6	man
	6	pres
	6	rep
	7	cod
	7	know
	7	rep
	8	admin
	8	hyp
	8	man
	9	admin
	9	eng
	9	sal
	10	admin
	10	eng
	10	sal

COMPANY

	companyid	companyname	representative	address	telephone
Þ	100	apple	carry	fake street	7536474
	101	google	billbo	Ballymun	9647653
	102	microsoft	gandalf	Detroit	7246457
	103	dell	ron	Los Angeles	8457453
	104	aspiron	penelope	San Francisco	1374564
	105	acer	justin	Germany	0785645
	106	lenovo	steven	London	7574643
	107	hp	paula	Essex	9568754
	108	dropbox	jenn	Dublin	8456453
	109	ebay	venessa	Galway	8346345
	110	amazon	helena	Berlin	7346342
	115	Smiths	barry	dggrsrdf	4574564
	116	Spitfire	shela	dfdrgdrfg	3463534

POSITIONS

	positionid	fkcompanynid	positionname
▶	80	100	programmer
	81	101	manager
	82	102	sales
	83	103	Programmer
	84	104	Manager
	85	105	Sales
	86	106	Programmer
	87	107	Manager
	88	108	Sales
	89	109	Programming
	90	110	Manager
	91	115	ceo
	92	115	sweepest

REQUIRESKILLS

	fkpositionid	fkskillid
Þ	80	cod
	80	know
	80	rep
	81	admin
	81	lead
	81	pres
	81	speak
	82	hyp
	82	sal
	83	eng
	83	man
	91	admin
	91	cod
	91	eng
	92	admin
	92	cod
	92	eng

SKILLS

	skillid	skillname
Þ	admin	administration
	cod	coding
	eng	english fluency
	hyp	hypnosis
	know	IT knowledge
	lead	leadership
	man	manipulation
	pres	presentation
	rep	IT repair
	sal	sales
	soc	social
	speak	speaking

INTERVIEWS

	interviewid	fk2_positionid	fk2_candidateid	interviewdate	outcome
Þ	1111	80	1	2016-01-04	success
	1112	80	2	2016-01-04	fail
	1113	80	3	2016-01-04	sucecss
	1114	80	4	2016-01-04	success
	1121	80	1	2016-01-10	fail
	1122	80	3	2016-01-10	success
	1123	80	4	2016-01-10	success
	1131	80	3	2016-01-22	success
	1132	80	4	2016-01-22	offer
	1141	80	3	2016-01-24	offer
	1211	81	4	2016-01-29	success
	1212	81	5	2016-01-29	success
	1213	81	6	2016-01-29	success
	1214	81	7	2016-02-12	fail
	1221	81	4	2016-02-12	success
	1222	81	5	2016-02-12	fail
	1223	81	6	2016-02-12	success
	1231	81	4	2016-02-25	fail
	1232	81	6	2016-02-25	offer
	1311	82	8	2016-03-02	fail
	1312	82	9	2016-03-02	success
	1313	82	10	2016-03-02	success
	1321	82	9	2016-03-02	success
	1322	82	10	2016-03-02	fail
	1331	82	9	2016-03-02	offer
	1411	90	10	2016-03-02	offer
	1511	89	8	2016-03-14	offer
	1611	87	7	2016-03-14	offer
	1711	86	6	2016-03-14	offer
	1811	85	5	2016-03-14	offer
	1911	83	1	2016-03-14	offer
	2011	84	2	2016-03-14	offer

HIRED

	fk3_candidateid	fk3_positionid
Þ	3	80
	4	80
	9	82
	1	83
	2	84
	5	85
	6	86
	7	87
	8	89
	10	90
	15	92

ASSUMPTIONS

- * All candidates and positions will be entered into the database with three skills related to them, no more. Achieved by:
- -ALLOWING MULTIPLE SKILLS= The skills which relate to the positions will be kept in a separate associative table called 'requiredskills' and the skills that relate to the candidate will be kept in a separate associative table called 'haveskills'
- -RESTRICT TO 3 SKILLS ON INPUT=The stored procedures which input the candidate/position will only have three input boxes relating to skills for that entry.
- *Interviews for apple programmer and google manager have finished and demonstrate multiple interviews for job and multiple interviews for candidates within set. The rest of the interviews are 1 each and they all get the job.

Achieved by:

- -ALLOWING MULTIPLE ROUNDS FOR POSITION = On interview table the user has manually inputed each interview so far
- -This is an associative table with candidateid, positionid as foreign keys to allow multiple rounds to occur relating to a position
- -The last 2 digits of the interviewid indicate how the rounds are broken up e.g. interviewid=1113 -> round 1 candidate 3
- *Candidates will be entered into the 'candidateshired' table if their last interview round in the interviews table is successful
- -There is a row in interviews table which records the outcome and if the last interview is a success 'offer' is entered and then the 'canidateshired' table is updated with the name of that candidate and the position that they have been hired for
- -The primary key in 'candidateshired' table will be candidateid(which is a foreign key) which will ensure each candidate can only have one row so therefore can only be conected to 1 position that they are hired for

