

**SICI-4030**  
**BASE DE DATOS**  
**EXAMEN PRÁCTICO 2**  
**Plantilla para entregar**

**INSTRUCCIONES :**

- I. Este trabajo se pasa a un documento en Word (INCLUIRLO) y se guarda en formato PDF (También incluirlo). Hay que dar *print-screen* a la matriz y al ERD que debe ser hecho en DIA. La matriz se puede hacer en EXCEL (*print-screen*) o dentro del mismo documento de Word(tabla). El formato del documento es: **Examen-1-NombreDelEstudiante.pdf**. RECUERDE NO PONER CARACTERES ESPECIALES EN EL NOMBRE DEL DOCUMENTO.
- II. **Diseño Físico** (incluya al menos 30 instancias por tabla) **(40 pts)**
- III. **DML** – Insertar Datos en las tablas. Tienen que ser los mismos datos del diseño Físico. únicamente en el orden correcto por niveles incluyendo los ***constraints*** (PK, FK, NOT NULL, etc.). Se deben incluir las instrucciones de DROP TABLE y la documentación requerida **(60 pts)**
- IV. Evidencia de la corrida. Sin esta parte, La parte de insertar datos no se acredita.

**Los documentos a entregar son:**

1. Matrix en Excel (**Exam-01-nombre-841-00-0000-Matrix.XLS**)
2. Matrix en PNG (*snipping*) (**Exam-01-nombre-841-00-0000-PNG.PNG**)
3. ERD en DIA (**Exam-01-nombre-841-00-0000-ERD.DIA**)
4. ERD en PNG (**Proy-01-nombre-841-00-0000-ERD.PNG**)
5. Documento de Word con todo lo requerido (**Exam-01-nombre-841-00-0000-TODO.DOCX**)
6. El mismo documento de Word en formato PDF (**Proy-01-nombre-841-00-0000-TODO.PDF**)

III. Diseño Físico (40 pts) Los datos no pueden ser compartidos. Las tablas van en plural. Usen el mismo diseño y pongan el título de la tabla centralizado antes del diseño físico. Son 10 instancias por tabla. Recuerde conectar el FK con el PK. Una tabla por página. No deje tablas en blanco en el documento. Plantilla de ejemplo:

DIRECCIONES

Nombre Columna	ID	Calle	Urbanización	Ciudad	Estado	Código Postal
Tipo de Key	PK					
Nulo/Único	NN, U	NN		NN	NN	NN
Datos de muestra.	1	712	Russell	Pensacola	FL	32590
	2	15085	Oriole	Los Angeles	CA	90040
	3	4	Becker	Erie	PA	16522
	4	86238	Graedel	Schenectady	NY	12305
	5	6	Sommers	Anchorage	AK	99522
	6	6	Bunker Hill	New York City	NY	10175
	7	862	Loomis	Fort Lauderdale	FL	33355
	8	372	Cordelia	Lakeland	FL	33811
	9	65	Becker	Philadelphia	PA	19146
	10	662	Ridge Oak	Van Nuys	CA	91406
	11	9	Hintze	San Francisco	CA	94121
	12	200	Arrowood	Pensacola	FL	32575
	13	8	Oneill	New York City	NY	10125
	14	40181	Logan	New Orleans	LA	70187
	15	7	Maryland	Fresno	CA	93704
	16	3985	Karstens	New York City	NY	10160
	17	82036	Melrose	Seattle	WA	98109
	18	2078	Acker	Rochester	MN	55905
	19	35	Graceland	Irvine	CA	92619
	20	2274	Kings	New York City	NY	10170
	21	6681	Springs	Oakland	CA	94622
	22	46	Duke	New Orleans	LA	70174
	23	381	Annamark	Tacoma	WA	98442
	24	76321	Arkansas	San Diego	CA	92153
	25	55944	Hoepker	Young America	MN	55564
	26	1115	Marcy	Fairbanks	AK	99709
	27	88257	Coolidge	Great Neck	NY	11024
	28	94	Northfield	Albany	NY	12247
	29	792	Eagan	Port Saint Lucie	FL	34985
	30	9277	Riverside	Garden Grove	CA	92645

ANIMALES

Nombre Columna	Código	Nombre	Descripción
Tipo de Key	PK		
Nulo/Único	NN, U	NN	NN
Datos de muestra.	A00	Gato	Tiene garras y necesita un scratchpad para poderse mantener las unas.
	A01	Perro	Energeticos y juguetones, por lo que necesita un espacio amplio.
	A02	Conejo	Larga y se arrastra, necesita ramas para poder desplazarse.
	A03	Serpiente	Necesita siempre tener heno disponible en su espacio.
	A04	Caballo	Son tenaces y hay que peinarle el cabello a menudo.
	A05	Vaca	Les gusta estar entre ellas mismas y se asustan fácilmente.
	A06	Cerdo	Les gusta rodarse en el lodo.
	A07	Guimo	No deben estar cerca de las serpientes porque pueden ser atacadas.
	A08	Raton	No pueden estar solas, deben estar alrededor de otras ratas.
	A09	Huron	Son curiosos, no deben estar alrededor de objetos peligrosos.
	B00	Cobaya	No deben estar alrededor de otros animales porque pueden ser agresivos.
	B01	Camaleon	Les gusta la soledad y se camuflasen utilizando su ambiente.
	B02	Lagarto	Son nerviosos, acercar con cuidado.
	B03	Arana	Son predadores, no se deben poner alrededor de otros insectos o animales.
	B04	Jirafa	Les gusta estar en grupo y son dociles.
	B05	Leon	Territoriales y agresivos, solo deben estar en grupos de su propia especie.
	B06	Oso	Si no estan en su propio nucleo familiar, deben estar solos.
	B07	Elefante	Son sociales pero deben aislarse cuando estan en apareamiento.
	B08	Tortuga	Pueden estar en soledad o en grupo, les gusta estar en el sol.
	B09	Cocodrilo	Predadores, aislados de todos los animales.
	C00	Iguana	Les gusta nadar y son nerviosos.
	C01	Delfin	Sociales y juguetones.
	C02	Ornitorrinco	Nocturnos, les gusta estar en grupo.
	C03	Gallina	Les gusta estar en grupo, son agresivos entre ellos.
	C04	Cabra	Tiene cuernos y es tenaz, por lo que no se debe de poner alrededor de otros.
	C05	Oveja	No se saben defender, no pueden estar cerca de animales agresivos.
	C06	Tigre	Predador agresivo, no deben estar cerca de otros animales.
	C07	Mono	Muy sociales, juguetones y carinosos.
	C08	Sapo	Viven en grupos y les gusta nadar.
	C09	Ciervo	Son tímidos y nerviosos, no se saben defender.

OBJETOS

Nombre Columna	ID	Nombre
Tipo de Key	PK	
Nulo/Único	NN, U	NN
Datos de muestra.	1A	Collar
	1B	Bowl de Agua
	1C	Bowl de comida
	1D	Sabana
	1E	Cama
	1F	Leash
	1G	Unas protectivas de gato
	1H	Esmalte de perro
	1I	Tinte de pelo para animales
	1J	Ropa de perro
	1K	Ropa de gato
	1L	Silla de montar caballo
	1M	Cortaunas
	1N	Scratchpad
	1O	Torre de gatoLitter de gato
	1P	Shampoo de perro
	1Q	Shampoo de gato
	1R	Despeluzante de gato
	1S	Maquina de recortar animales
	1T	Peine
	1U	Piscina de animales
	1V	Hueso
	1W	Juguete de gato
	1X	Juguete de perro
	1Y	Gafas de perro
	1Z	Silla de rueda de perro
	2A	Cono para perro
	2B	Cono para gato
	2C	Zapatos de perro
	2D	Zapatos de gato

CONDICIONES

Nombre Columna	ID	Nombre	Descripción
Tipo de Key	PK		
Nulo/Único	NN, U	NN	NN
Datos de muestra.	AB012	Epilepsia	Convulsiones y confusion al momento.
	AB013	Leucemia	Cancer de sangre.
	AC012	Eczema	Inflamacion de la piel y ardor en la misma.
	AC013	Fallo Renal	Propenso a deshidratacion. Dieta especial.
	RT876	Pulgas	Causa inflamacion y perdida de pelo.
	HY895	Garrapatas	Causan malestar intenso y es posible pasarlo a los humanos.
	WD326	Artritis	Inflamacion en las articulaciones y dolor al momento de mover.
	TR457	Parasitos	Causan vomitos, diarrea y dolor abdominal.
	YP988	Moquillo	Parecido al sarampión, necesita antibioticos.
	JA349	Leptospirosis	Causada por una bacteria, necesita antibioticos.
	QW023	Hepatitis	Muy contagiosa, tratamiento intensivo.
	TE002	Parvovirus	Enfermedad virica muy contagiosa.
	WS345	Rabia	Enfermedad mortal, se evita por vacunacion.
	GH641	Brucelosis	Causada por una bacteria, causa esterilidad y aborto espontaneo.
	FD567	Lyme	Causada por una bacteria, se transmite por una garrapata infectada.
	AS578	Traqueobronquitis	Enfermedad respiratoria muy contagiosa.
	RF895	Coronavirus	Generalmente leve pero muy contagiosa, necesita antibioticos.
	UD765	Toxoplasmosis	No presenta sintomas, necesita antibioticos.
	DQ865	Otitis	Infeccion del conducto auditivo.
	QD114	Conjuntivitis	Infalamcion de la membrana conjuntiva.
	WQ345	Alergia cutanea	Es necesario administrar algun antihistaminico.
	QE865	Gastritis aguda	Inflamacion de la pared del estomago, causa vomitos graves.
	AW235	Diarrea	Deposicion de heces liquidas o informes.
	WE456	Leishmaniosis	Enfermedad parasitaria grave, incluso mortal.
	HD333	Sarna demodectica	Fauna habitual de la piel.
	XC534	Mastitis	Endurecimiento de las glandulas mamarias.
	SF477	Diabetes	Falta de insulina o respuesta inadecuada de esta.
	FP864	Tina	Causada por un hongo que infecta la piel y deja calvas.
	WZ768	Gusano del corazon	Parasito que puede instalarse en el corazon y arterias pulmonares.
	HR384	Tos de las perreras	Comparable con un resfriado de pecho.

ENCASILLADOS

Nombre Columna	ID	Descripción
Tipo de Key	PK	
Nulo/Único	NN, U	NN
Datos de muestra.	313	Cama para perro.
	314	Cama y scratchpad para gato.
	315	Seccion de la finca para caballo.
	316	Ramas para serpiente trepar.
	327	Cama con heno para conejo.
	328	Cama para cabra.
	329	Tanque para delfin.
	330	Seccion de la finca para vaca.
	331	Jaula para guimo con hamster wheel.
	332	Jaula para raton.
	333	Jaula para huron
	334	Jaula para perros grandes.
	335	Jaula solitaria para los perros agresivos.
	336	Jaula solitaria para los gatos antisociales.
	337	Jaula grande para las gatas madres y su gatitos hijos.
	338	Seccion de la finca para cerdos.
	339	Jaula con ramas para camaleon.
	340	Tanque de agua salada para delfines.
	341	Seccion de la finca para gallinas.
	342	Envoltura con arena y agua salada para tortugas
	343	Seccion del area desertico para tigres.
	344	Seccion del area desertico para elefantes.
	345	Seccion del area desertico para jirafas.
	346	Seccion del area tropical para monos.
	347	Seccion del area tropical para iguanas.
	348	Seccion del area tropical para sapos.
	349	Seccion del area de bosque para ciervos.
	350	Seccion del area de bosque para osos.
	351	Jaula para cobaya.
	352	Seccion del area de pantano para cocodrilo.

DUENOS

Nombre Columna	ID	Nombre()				Celular	Correo Electrónico	ID Dirección
		Primer Nombre	Inicial	Apellido Paterno	Apellido Materno			
Tipo de Key	PK							FK
Nulo/Único	NN, U	NN		NN		NN	NN	NN
Datos de muestra.	12311	Jorgan	E	Leyre	Callery	7876756677	jcallery0@baidu.com	1
	12312	Pegeen	G	Cheese	Charlo	7875646633	pcharlo1@dion.ne.jp	2
	12313	Selby	C	Gouldsmith	Scampion	7877674432	sscampion2@google.it	3
	12314	Correna	T	Fehner	Pirolini	9399864735	cpirolini3@google.de	4
	12315	Damita	H	Yare	Gilbank	7878926743	dgilbank4@pen.io	5
	12316	Jerrold	J	O'Lynn	Sirmon	7878267472	jsirmon5@bluehost.com	6
	12317	Dorri	W	Simeon	Hulks	9399836729	dhulks6@studiopress.com	7
	12318	Peadar	H	Prentice	Tull	9390017362	ptull7@yahoo.co.jp	8
	12319	Bella	F	Romke	Morris	7879277193	bmorris8@prweb.com	9
	12320	Mahala	W	Houldcroft	MacCart	7879278267	mmaccart9@admin.ch	10
	12321	Deeyn	F	Saipy	Milton-White	7879871242	dmiltonwhitea@quantcast.com	11
	12322	Odille	S	Harmon	Ferrel	7870872663	oferrelib@lycos.com	12
	12323	Elyssa	D	Massimo	Merigeau	7878937932	emerigeauc@slashdot.org	13
	12324	Emlyn	Q	Clark	Eglese	9394235754	eeglesed@smh.com.au	14
	12325	Ola	E	Brentnall	Cleen	7874574425	ocleene@canalblog.com	15
	12326	Zedekiah	E	Scarre	Boldock	7873567853	zboldockf@bandcamp.com	1
	12327	Roshelle	W	Humbee	Miquelet	9395787654	rmiqueletg@cbslocal.com	2
	12328	Hattie	T	Puddle	Wrennall	7870987623	hwrennallh@tumblr.com	3
	12329	Carl	E	Byfford	Dyke	7878762452	cdykei@latimes.com	4
	12330	Binny	E	Thomelin	Catto	7877345674	bcattoj@eventbrite.com	5
	12331	Hedy	M	Elliss	Fihelly	7877666631	hfihellyk@macromedia.com	6
	12332	Mill	Q	Bowkley	Culvey	7870872224	mculveyl@shop-pro.jp	7
	12333	Ivett	A	Moden	Marr	7877654245	imarrm@furl.net	8
	12334	Gerri	B	Jump	Blown	9390764214	gblownnn@bravesites.com	9
	12335	Pamelina	D	Slainey	Spencook	9397454535	pspencooko@bigcartel.com	10
	12336	Garwin	W	Waite	Faulkes	7875532654	gfaulkesp@intel.com	11
	12336	Phyllis	I	Bilton	Pedican	7870876245	ppedicanq@jigsy.com	12
	12337	Cristie	F	Juanico	Pedroni	7878734135	cpedronir@yahoo.com	13
	12338	Case	H	Hallums	Guy	7874346876	cguys@mtv.com	14
	12339	Wald	C	Thorwarth	Alleburton	7870876424	walleburtont@dell.com	15

COMIDAS

Nombre Columna	Numero	Nombre	Unidad de Medida	ID Dueño
Tipo de Key	PK			FK
Nulo/Único	NN, U	NN	NN	NN
Datos de muestra.	886523	Meow Mix	25 lb	12311
	886524	Friskies Comida Seca	15 lb	12312
	886525	Friskies Treats	1 bolsa	12313
	886526	Purina One	10 lb	12314
	886527	Purina Pro Plan	18 lb	12315
	886528	Royal Canine Comida Mojada	2 latas	12316
	886529	ACANA Comida Mojada	3 latas	12317
	886530	Meow Mix	20 lb	12318
	886531	Royal Canine Comida Mojada	4 latas	12319
	886532	Friskies Comida Mojada	1 lata	12320
	886533	Purina One	15 lb	12321
	886534	Meow Mix	25 lb	12322
	886535	Friskies Comida Seca	15 lb	12323
	886536	Friskies Treats	1 bolsa	12324
	886537	Purina One	10 lb	12325
	886538	Blue Comida en Seco	20 lb	12326
	886539	Arroz y pollo	3 lb	12327
	886540	Vegetales hervidos	1 lb	12328
	886541	Carne Cruda	5 lb	12329
	886542	Purina Pro Plan	10 lb	12330
	886543	Cesar Classics Comida Mojada	3 envases	12331
	886544	Milk Bone Treats	1 caja	12332
	886545	The Honest Kitchen Comida Mojada	3 envases	12333
	886546	ACANA Comida Seca	2 lb	12334
	886547	Blue Buffalo Comida en Seco	10 lb	12335
	886548	Arroz y pollo	3 lb	12336
	886549	Vegetales hervidos	1 lb	12336
	886550	Carne Cruda	5 lb	12337
	886551	Purina Pro Plan	10 lb	12338
	886552	Royal Canine Comida Seca	5 lb	12339



OFICINAS

Nombre Columna	ID	Teléfono	Correo Electrónico	ID Direccion
Tipo de Key	PK			FK
Nulo/Único	NN, U	NN	NN	NN
Datos de muestra.	7865	4921817874	cmothersole12@chron.com	16
	7866	7124264823	mportail13@weather.com	17
	7867	7624452717	hcoathup14@home.pl	18
	7868	2834828857	rsutch15@oakley.com	19
	7869	6288331615	jcalderon16@about.me	20
	7870	5353681232	dmcanellye17@uiuc.edu	21
	7871	8939583320	droderigo18@fastcompany.com	22
	7872	7938584322	evanin19@gmpg.org	23
	7873	3945818644	alejeune1a@seattletimes.com	24
	7874	6181522638	bjermin1b@google.it	25
	7875	4899297003	hchestnutt1c@mysql.com	26
	7876	9684249416	ischuchmacher1d@de.vu	27
	7877	4207668874	naxelbey1e@house.gov	28
	7878	1208857499	thaistwell1f@nih.gov	29
	7879	2101444349	vabbay1g@ox.ac.uk	30
	7880	3165921401	mtrengrove1h@wikimedia.org	16
	7881	5233838620	byanson1i@bing.com	17
	7882	1638139625	pjoynson1j@aboutads.info	18
	7883	4986564011	hbroinlich1k@sitemeter.com	19
	7884	5995117580	varlott1l@canalblog.com	20
	7885	6144424848	hbuckleigh1m@tiny.cc	21
	7886	4524778599	jcuppitt1n@about.me	22
	7887	1111976892	phouston1o@123-reg.co.uk	23
	7888	8152994430	ctaylo1p@flavors.me	24
	7889	3706493199	dhanner1q@foxnews.com	25
	7890	2116348422	blorenzini1r@nydailynews.com	26
	7891	1661635203	gcarolan1s@sciencedirect.com	27
	7892	8686013951	dschrinel1t@virginia.edu	28
	7893	1919466178	ibrownell1u@angelfire.com	29
	7894	5526311340	iveltman1v@miitbeian.gov.cn	30

CLINICAS

Nombre Columna	ID	Teléfono	Correo Electrónico	ID Direccion
Tipo de Key	PK			FK
Nulo/Único	NN, U	NN	NN	NN
Datos de muestra.	3228	5203588600	gbroz2q@alibaba.com	16
	3229	5453373524	jmuriel2r@tripadvisor.com	17
	3230	2811519751	mralphs2s@uol.com.br	18
	3231	6017053928	egiorgetti2t@smh.com.au	19
	3232	7748219496	mautin2u@storify.com	20
	3233	7936439279	gwestbury2v@ted.com	21
	3234	3788598848	ecraister2w@bbb.org	22
	3235	9256753186	mtorrisi2x@noaa.gov	23
	3236	1814669361	acalltone2y@bbc.co.uk	24
	3237	8674532225	tgrendon2z@purevolume.com	25
	3238	4643741359	ashervil30@newsvine.com	26
	3239	3802663782	lmacgillivrie31@pbs.org	27
	3240	8527218744	plohmeier32@rambler.ru	28
	3241	8452250787	aproger33@si.edu	29
	3242	7332825132	mkellegher34@bloomberg.com	30
	3243	5152191049	plusher35@engadget.com	16
	3244	2075538147	fleele36@uiuc.edu	17
	3245	8836138672	tcartmael37@4shared.com	18
	3246	2353114213	dbrand38@cyberchimps.com	19
	3247	4774903438	gbrafferton39@ezinearticles.com	20
	3248	1756543871	ffoulgham3a@stumbleupon.com	21
	3249	5384164159	krappoport3b@ucoz.ru	22
	3250	8126772863	rirnys3c@twitpic.com	23
	3251	5956823293	cvearncomb3d@google.co.uk	24
	3252	1391023724	ogrose3e@harvard.edu	25
	3253	5782042072	cdawks3f@twitpic.com	26
	3254	8021231672	bantrum3g@xinhuanet.com	27
	3255	1281531682	pkaemena3h@newsvine.com	28
	3256	2847166433	etaborre3i@reddit.com	29
	3257	3467122737	cleguey3j@chicagotribune.com	30

DEPARTAMENTOS

Nombre Columna	Codigo	Nombre	Descripción	ID Oficina
Tipo de Key	PK			FK
Nulo/Único	NN, U	NN	NN	NN
Datos de muestra.	Z0	Recursos Humanos	Encargados de administrar los asuntos interpersonales de la compania.	7865
	Z1	Finanzas	Aseguran que el aspecto financiero de la compania se esta administrando de la manera correcta.	7866
	Z2	Mercadeo	Intentan atraer a mas clientes mediante anuncios.	7867
	Z3	Ejecutivo	Toman decisiones importantes.	7868
	Z4	Manejo de Operaciones	Regulan operaciones.	7869
	Z5	Information Technology	Trabajan con la tecnología de la compania.	7870
	Z6	Junta de Directores	Toman las decisiones mas importantes.	7871
	Z7	Investigacion	Completan investigaciones.	7872
	Z8	Manufactura	Se encargan de hacer los productos que necesitan en las clínicas.	7873
	Z9	Servicio al Cliente	Ayudan a los clientes por si algún problema ocurre.	7874
	Y0	Legal	Manejan los aspectos legales de la compania.	7875
	Y1	Entrenamiento	Le dan entrenamiento a los empleados nuevos.	7876
	Y2	Seguridad	Manejan la seguridad de la compania.	7877
	Y3	Administración General	Administran los aspectos de la clinica.	7878
	Y4	Ventas	Venden comida y objetos de animales.	7879
	Y5	Distribucion	Distribuyen los artículos de venta y meciamentos a las distinta clinicas.	7880
	Y6	Compras	Compran los materiales necesarios para nuestras clinica.	7881
	Y7	Calidad	Aseguran que la clínica este haciendo el trabajo de manera correcta.	7882
	Y8	Ingenieria	Encargados de crear nuevas herramientas para el uso de las clinicas.	7883
	Y9	Farmaceutica	Desarrollan y administran los medicamentos para los pacientes.	7884
	X0	Redes Sociales	Encargados de aumentar nuestra clientela mediante el uso de redes sociales.	7885
	X1	Mantenimiento	Se encargan de limpiar las facilidades.	7886
	X2	Diseno de Productos	Disenan nuevas comidas y objetos para nuestros pacientes.	7887
	X3	Logistica	Analizan nuestro progreso y hacen recomendaciones.	7871
	X4	Transportacion	Encargados de la transportacion de bienes entre nuestras facilidades.	7872
	X5	Relaciones Publicas	Regulan la relación que tiene la compania con su clientela.	7873
	X6	Inversiones	Encargados de los stocks de la compania.	7874
	X7	Auditoria	Regulan que la compania siga reglamentos y leyes.	7875
	X8	Diseño	Disenan nuevos productos, trabajan mano a mano con ingenieria.	7876
	X9	Contabilidad	Llevan cuenta del flujo de dinero en la compania.	7877

EMPLEADOS																
Nombre Columna	Numero	Nombre()				Tipo	Salario Mensual	Facturacion Contrato	Numero de contrato	Salario por hora	Horas trabajadas	Fecha de Nacimiento	Correo Electrónico	ID Dirección	Código Dept.	ID Clínica
		Primer Nombre	Inicial	Apellido Paterno	Apellido Materno											
Tipo de Key	PK													FK1	FK2	FK3
Nulo/Único	NN, U	NN		NN		NN						NN	NN	NN	NN	NN
Datos de muestra.	86734	Mortimer	J	Goundry	Robles	C		200.00	1234			12-05-1978	mgoundry0@dyndns.org	1	Z0	
	86735	Daisey		Lissandrini	Cheese	C		650.00	1235			15-09-1950	dlissandrini1@usa.gov	2	Z1	
	86736	Alwin	D	Beavon	Gouldsmith	C		850.00	1236			26-07-1961	abeavon2@360.cn	3	Z2	
	86737	Augusto	G	Scatchard	Fehner	C		999.99	1237			05-08-1987	ascatchard3@diigo.com	4	Z3	
	86738	Faber	W	Scamerdine	Yare	C		325.00	1238			30-09-2000	fscamdine4@apache.org	5	Z4	
	86739	Bordy		Boycott	OLynn	C		765.50	1239			01-05-1996	bboycott5@mlb.com	6	Z5	
	86740	Melosa	Y	Lupson	Simeon	C		230.00	1240			26-07-1952	mlupson6@vista.com	7	Z6	
	86741	Roland	K	Dunham	Prentice	C		768.00	1241			13-08-2001	rdunham7@toplist.cz	8	Z7	
	86742	Idelle	W	Mart	Romke	C		433.33	1242			18-02-1971	imart8@livejournal.com	9	Z8	
	86743	Barron	R	Endricci	Houldcroft	C		545.54	1243			31-08-1967	bendricci9@4shared.com	10	Z9	
	86744	Jayson	F	Quaife	Saipy	A	3450.00					13-06-1984	jquaifea@deviantart.com	11	Y0	
	86745	Noelani	S	Lempenny	Harmon	A	5600.95					19-04-1991	nlempennyb@hc.com	12	Y1	
	86746	Bianka		Skepper	Massimo	A	2098.00					25-05-1981	bskepper@dede.com	13	Y2	
	86747	Nappy	P	Deinhard	Clark	A	4500.00					28-11-1975	ndeinhardd@ning.com	14	Y3	
	86748	Gery	T	Brockie	Brentnall	A	1500.00					01-12-1970	gbrockiee@name.com	15	Y4	
	86749	Angie	E	Stickney	Scarre	A	7650.50					28-07-1950	astickneyf@parall.com	1		16
	86750	Theodoric	F	Sheirlaw	Humbee	A	2500.00					27-06-1954	tsheirlawg@huff.com	2		17
	86751	Shelbi	M	Woolger	Puddle	A	2500.00					16-10-2002	swoolgerh@yellow.com	3		18
	86752	Celka	F	Golde	Byfford	A	4322.00					20-10-1984	cgoldei@unblog.fr	4		19
	86753	Costanza	I	Robardet	Thomelin	A	6000.00					16-03-1994	crobardetj@usgs.gov	5		20
	86754	Lauren	F	Findley	Elliss	H				7.25	39	19-11-1969	lfindleyk@devhub.com	6		21
	86755	Courtney	P	Boultwood	Bowkley	H				8.50	38	06-08-1989	cboultwoodl@mit.edu	7		22
	86756	Lazare	P	Cammock	Moden	H				15.00	37	19-10-1987	lcammockm@ed.gov	8		23
	86757	Hilliard	T	Staggs	Jump	H				10.00	45	15-09-1954	hstaggsn@forbes.com	9		24
	86758	Cello	R	Dyke	Slainey	H				27.00	24	28-08-1979	cdykeo@amazon.com	10		25
	86759	Daisi	A	Blaine	Waite	H				13.00	23	17-08-1988	dblaine@google.co.jp	11		26
	86760	Zechariah	H	Dumper	Bilton	H				14.50	65	29-12-1999	zdumperq@a8.net	12		27
	86761	Idette	E	Burgyn	Juanico	H				25.00	36	03-11-1981	iburgynr@unblog.fr	13		28
	86762	Emmalee	P	Rissen	Hallums	H				18.00	80	01-07-1965	erissens@loc.gov	14		29
	86763	Farleigh	K	Borsi	Thorwarth	H				32.00	40	25-10-1976	fborsit@google.co.uk	15		30

MASCOTAS

Nombre Columna	Numero	Nombre	Fecha de Nacimiento	Raza	ID Dueno	ID Animal
Tipo de Key	PK				FK1	FK2
Nulo/Único	NN, U	NN	NN	NN	NN	NN
Datos de muestra.	610873	Luna	29-10-2004	Maine Coon	12311	A00
	610874	Firulais	16-12-2009	Persa	12312	A00
	610875	Coco	21-11-2005	Esfinge	12313	A00
	610876	Jellybean	24-03-2022	Siames	12314	A00
	610877	Jaruby	13-02-2006	Bengali	12315	A00
	610878	Nio	13-07-2007	Bosque de Noruega	12316	A00
	610879	Shiny	26-05-2010	Siberiano	12317	A00
	610880	Nico	24-06-2006	Azul Ruso	12318	A00
	610881	Safira	26-10-2004	Ragdoll	12319	A00
	610882	Dario	29-08-2017	British Shorthair	12320	A00
	610883	Lola	26-12-2020	Oriental	12321	A00
	610884	Petunia	17-08-2020	Birmano	12322	A00
	610885	Mia	27-12-2012	Angora Turco	12323	A00
	610886	Nana	29-07-2004	Van Turco	12324	A00
	610887	Simba	27-09-2020	Salchicha	12325	A01
	610888	Thor	19-07-2009	Pastor Aleman	12326	A01
	610889	Max	13-01-2019	Shitzu	12327	A01
	610890	Rocky	28-02-2015	Shiba Inu	12328	A01
	610891	Leo	30-04-2010	Doberman	12329	A01
	610892	Lucas	18-10-2014	Basset hound	12330	A01
	610893	Bruno	29-03-2016	Border collie	12331	A01
	610894	Noa	19-10-2016	Boxer	12332	A01
	610895	Kira	23-05-2017	Poodle	12333	A01
	610896	Bimbo	16-06-2011	Golden Retriever	12334	A01
	610897	Choco	22-03-2014	Chihuahua	12335	A01
	610898	Lima	30-11-2020	Pug	12336	A01
	610899	Sushi	20-07-2020	Pomeranian	12336	A01
	610900	Trufa	22-09-2019	Maltese	12337	A01
	610901	Oreo	17-06-2014	Rottweiler	12338	A01
	610902	Loki	29-10-2020	Sato	12339	A01

RESERVACIONES

Nombre Columna	Numero	Fecha Inicial	Fecha Final	Numero Mascota	ID Encasillado
Tipo de Key	PK			FK1	FK2
Nulo/Único	NN, U	NN		NN	NN
Datos de muestra.	1	21-01-2022	18-02-2022	610873	314
	2	26-01-2022	27-03-2022	610874	314
	3	31-01-2022		610875	314
	4	14-01-2022	30-04-2022	610876	336
	5	02-01-2022	15-05-2022	610877	336
	6	04-01-2022	17-04-2022	610878	336
	7	29-01-2022	14-02-2022	610879	337
	8	15-01-2022	20-04-2022	610880	337
	9	13-01-2022	15-05-2022	610881	337
	10	03-01-2022	16-05-2022	610882	314
	11	15-01-2022		610883	336
	12	09-01-2022	30-04-2022	610884	336
	13	16-01-2022	15-05-2022	610885	336
	14	05-01-2022	17-04-2022	610886	337
	15	06-01-2022	14-03-2022	610887	337
	16	27-01-2022	22-04-2022	610888	313
	17	04-01-2022	05-04-2022	610889	313
	18	04-01-2022	05-05-2022	610890	313
	19	01-01-2022	30-04-2022	610891	334
	20	11-01-2022		610892	334
	21	24-01-2022	28-02-2022	610893	334
	22	22-01-2022	24-03-2022	610894	335
	23	02-01-2022	26-03-2022	610895	335
	24	06-01-2022	27-03-2022	610896	335
	25	23-01-2022	13-03-2022	610897	334
	26	04-01-2022	25-02-2022	610898	335
	27	22-01-2022	14-02-2022	610899	313
	28	07-01-2022	20-04-2022	610900	313
	29	20-01-2022	15-05-2022	610901	334
	30	28-01-2022	20-05-2022	610902	313

RECORDS

Nombre Columna	ID Clinica	Numero Mascota
Tipo de Key	PK, FK1	PK, FK2
Nulo/Único	NN	NN
Datos de muestra.	3228	610873
	3229	610874
	3230	610875
	3231	610876
	3232	610877
	3233	610878
	3234	610879
	3235	610880
	3236	610881
	3237	610882
	3238	610883
	3239	610884
	3240	610885
	3241	610886
	3242	610887
	3243	610888
	3244	610889
	3245	610890
	3246	610891
	3247	610892
	3248	610893
	3249	610894
	3250	610895
	3251	610896
	3252	610897
	3253	610898
	3254	610899
	3255	610900
	3256	610901
	3257	610902

MASCOTA-COMIDAS

Nombre Columna	Numero Mascota	Numero Comida
Tipo de Key	PK, FK1	PK, FK2
Nulo/Único	NN	NN
Datos de muestra.	610873	886523
	610874	886524
	610875	886525
	610876	886526
	610877	886527
	610878	886528
	610879	886529
	610880	886530
	610881	886531
	610882	886532
	610883	886533
	610884	886534
	610885	886535
	610886	886536
	610887	886537
	610888	886538
	610889	886539
	610890	886540
	610891	886541
	610892	886542
	610893	886543
	610894	886544
	610895	886545
	610896	886546
	610897	886547
	610898	886548
	610899	886549
	610900	886550
	610901	886551
	610902	886552



MASCOTA-OBJETOS

Nombre Columna	Numero Mascota	ID Objeto
Tipo de Key	PK, FK1	PK, FK2
Nulo/Único	NN	NN
Datos de muestra.	610873	1A
	610874	1B
	610875	1C
	610876	1D
	610877	1E
	610878	1F
	610879	1S
	610880	1T
	610881	1A
	610882	1B
	610883	1C
	610884	1D
	610885	1E
	610886	1F
	610887	1S
	610888	1T
	610889	1D
	610890	1E
	610891	1F
	610892	1S
	610893	1T
	610894	1A
	610895	1B
	610896	1C
	610897	1D
	610898	1E
	610899	1F
	610900	1D
	610901	1E
	610902	1F

MASCOTA-CONDICIONES

Nombre Columna	Numero Mascota	ID Condición
Tipo de Key	PK, FK1	PK, FK2
Nulo/Único	NN	NN
Datos de muestra.	610873	AB012
	610874	AB013
	610875	AC012
	610876	AC013
	610877	RT876
	610878	HY895
	610879	WD326
	610880	TR457
	610881	YP988
	610882	JA349
	610883	QW023
	610884	TE002
	610885	WS345
	610886	GH641
	610887	FD567
	610888	AS578
	610889	RF895
	610890	UD765
	610891	DQ865
	610892	QD114
	610893	WQ345
	610894	QE865
	610895	AW235
	610896	WE456
	610897	HD333
	610898	XC534
	610899	SF477
	610900	FP864
	610901	WZ768
	610902	HR384

#### IV. DDL – Creación Tablas (30 pts) incluir código y *printscreen* de la corrida.

Recuerde incluir los **DROP TABLES**, el comando que muestra la fecha y hora y comando que muestra el *username* antes de cualquier instrucción del script. Se tiene que llenar la documentación que se solicita:

**Codigo solamente** (se tiene que modificar e incluir el código completo aquí. No se puede dejar la plantilla original):

```
REM *****
REM *   Author       : Jillian S. Alvarez Sanlley          *
REM *   Num.Est      : 841-19-6451                        *
REM *   Curso        : SICI-4030                          *
REM *   Sección      : KR1                                 *
REM *   Source       : Plantilla-SQL-Script-Documentado.sql *
REM *   Comentario   : Este script es para el Proyecto Final. *
REM *****
REM *****
REM *   EL SET ECHO ON permite ver la corrida en pantalla.  *
REM *****
SET ECHO ON
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM *   Aquí se borran las tablas en caso de tener que volver a *
REM *   crearlas, esta sub dividio por niveles.                *
REM *****
DROP TABLE mascotaobjeto;
DROP TABLE mascotacondicion;
DROP TABLE mascotacomida;
DROP TABLE record;
DROP TABLE reservacion;
DROP TABLE mascota;
DROP TABLE empleado;
DROP TABLE clinica;
DROP TABLE departamento;
DROP TABLE oficina;
DROP TABLE comida;
DROP TABLE dueno;
DROP TABLE encasillado;
DROP TABLE condicion;
DROP TABLE objeto;
DROP TABLE animal;
DROP TABLE direccion;
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
```

```

REM * Creacion tabla direccion nivel-1
REM *****
CREATE TABLE direccion (
    id_pk          VARCHAR2(4)    NOT NULL,
    calle          VARCHAR2(10)   NOT NULL,
    urbanizacion   VARCHAR2(25),
    ciudad         VARCHAR2(15)   NOT NULL,
    estado         CHAR(2)        NOT NULL,
    codigo_postal  CHAR(5)        NOT NULL,
CONSTRAINT direccion_id_pk PRIMARY KEY (id_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del
REM * usuario de la cuenta y la fecha del sistema. Esto es
REM * requerido para crear o eliminar tablas y otros objetos.
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla animal nivel-1
REM *****
CREATE TABLE animal (
    codigo_pk      VARCHAR2(3)    NOT NULL,
    nombre         VARCHAR2(25)   NOT NULL,
    descripcion     VARCHAR2(99)  NOT NULL,
CONSTRAINT animal_codigo_pk PRIMARY KEY (codigo_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del
REM * usuario de la cuenta y la fecha del sistema. Esto es
REM * requerido para crear o eliminar tablas y otros objetos.
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla objeto nivel-1
REM *****
CREATE TABLE objeto (
    id_pk          VARCHAR2(2)    NOT NULL,
    nombre         VARCHAR2(15)   NOT NULL,
CONSTRAINT objeto_id_pk PRIMARY KEY (id_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del
REM * usuario de la cuenta y la fecha del sistema. Esto es
REM * requerido para crear o eliminar tablas y otros objetos.
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla condicion nivel-1
REM *****
CREATE TABLE condicion (
    id_pk          VARCHAR2(5)    NOT NULL,
    nombre         VARCHAR2(15)   NOT NULL,
    descripcion     VARCHAR2(99)  NOT NULL,
CONSTRAINT condicion_id_pk PRIMARY KEY (id_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del
REM * usuario de la cuenta y la fecha del sistema. Esto es
REM * requerido para crear o eliminar tablas y otros objetos.
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;

```

```

REM *****
REM * Creacion tabla encasillado nivel-1 *
REM *****
CREATE TABLE encasillado (
    id_pk                NUMBER(3)    NOT NULL,
    descripcion          VARCHAR2(99) NOT NULL,
    CONSTRAINT encasillado_id_pk PRIMARY KEY (id_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla dueno nivel-2 *
REM *****
CREATE TABLE dueno (
    id_pk                NUMBER(5)    NOT NULL,
    primer_nombre        VARCHAR2(20) NOT NULL,
    inicial              CHAR(1),
    apellido_paterno     VARCHAR2(20) NOT NULL,
    apellido_materno     VARCHAR2(20),
    celular               CHAR(10)    NOT NULL,
    correo_electronico    VARCHAR2(30) NOT NULL,
    id_direccion_fk      VARCHAR2(4)  NOT NULL,
    CONSTRAINT dueno_iddireccion_fk FOREIGN KEY (id_direccion_fk)
        REFERENCES direccion(id_pk),
    CONSTRAINT dueno_id_pk PRIMARY KEY (id_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla comida nivel-2 *
REM *****
CREATE TABLE comida (
    num_pk               VARCHAR2(5)  NOT NULL,
    nombre               VARCHAR2(25) NOT NULL,
    unidad_medida        VARCHAR2(15) NOT NULL,
    id_dueno_fk          NUMBER(7,0)  NOT NULL,
    CONSTRAINT comida_iddueno_fk FOREIGN KEY (id_dueno_fk)
        REFERENCES dueno(id_pk),
    CONSTRAINT comida_num_pk PRIMARY KEY (num_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla oficina nivel-2 *
REM *****
CREATE TABLE oficina (
    id_pk                NUMBER(4)    NOT NULL,
    telefono              CHAR(10)    NOT NULL,
    correo_electronico    VARCHAR2(30) NOT NULL,
    id_direccion_fk      VARCHAR2(4)  NOT NULL,

```

```

CONSTRAINT oficina_iddireccion_fk FOREIGN KEY (id_direccion_fk)
REFERENCES direccion(id_pk),
CONSTRAINT oficina_id_pk PRIMARY KEY (id_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla clinica nivel-2 *
REM *****
CREATE TABLE clinica (
    id_pk NUMBER(4) NOT NULL,
    telefono CHAR(10) NOT NULL,
    correo_electronico VARCHAR2(30) NOT NULL,
    id_direccion_fk VARCHAR2(4) NOT NULL,
CONSTRAINT clinica_iddireccion_fk FOREIGN KEY (id_direccion_fk)
REFERENCES direccion(id_pk),
CONSTRAINT clinica_id_pk PRIMARY KEY (id_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla departamento nivel-2 *
REM *****
CREATE TABLE departamento (
    codigo_pk CHAR(2) NOT NULL,
    nombre VARCHAR2(20) NOT NULL,
    descripcion VARCHAR2(99) NOT NULL,
    id_oficina_fk NUMBER(4) NOT NULL,
CONSTRAINT departamento_idoficina_fk FOREIGN KEY (id_oficina_fk)
REFERENCES oficina(id_pk),
CONSTRAINT departamento_codigo_pk PRIMARY KEY (codigo_pk));CONSTRAINT
departamento_codigo_pk PRIMARY KEY (codigo_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla empleado nivel-3 *
REM *****
CREATE TABLE empleado (
    num_pk NUMBER(5) NOT NULL,
    primer_nombre VARCHAR2(20) NOT NULL,
    inicial CHAR(1),
    apellido_paterno VARCHAR2(20) NOT NULL,
    apellido_materno VARCHAR2(20),
    tipo CHAR(1),
    salario_mensual NUMBER(7,2),
    num_contrato NUMBER(4),
    tarifa_contrato NUMBER(5,2),
    salario_x_hora NUMBER(4,2),
    horas_trabajadas NUMBER(2),

```

```

        fecha_nacimiento          DATE          NOT NULL,
        correo_electronico        VARCHAR2(30)    NOT NULL,
        id_direccion_fk1          VARCHAR2(4)     NOT NULL,
        codigo_departamento_fk2  CHAR(2),
        id_clinica_fk3            NUMBER(4),
CHECK ((tipo) IN ('C', 'H', 'A')),
CONSTRAINT empleado_iddireccion_fk1 FOREIGN KEY (id_direccion_fk1)
        REFERENCES direccion(id_pk),
CONSTRAINT empleado_codigodepartamento_fk2 FOREIGN KEY (codigo_departamento_fk2)
        REFERENCES departamento(codigo_pk),
CONSTRAINT empleado_idclinica_fk3 FOREIGN KEY (id_clinica_fk3)
        REFERENCES clinica(id_pk),
CONSTRAINT empleado_num_pk PRIMARY KEY (num_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla mascota nivel-3 *
REM *****
CREATE TABLE mascota (
        num_pk          NUMBER(6)    NOT NULL,
        nombre          VARCHAR2(15) NOT NULL,
        fecha_nacimiento DATE          NOT NULL,
        raza            VARCHAR2(20) NOT NULL,
        id_dueno_fk1     NUMBER(7,0)  NOT NULL,
        codigo_animal_fk2 VARCHAR2(4)  NOT NULL,
CONSTRAINT mascota_iddueno_fk FOREIGN KEY (id_dueno_fk1)
        REFERENCES dueno(id_pk),
CONSTRAINT mascota_codanimal_fk FOREIGN KEY (codigo_animal_fk2)
        REFERENCES animal(codigo_pk),
CONSTRAINT mascota_num_pk PRIMARY KEY (num_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla reservacion nivel-3 *
REM *****
CREATE TABLE reservacion (
        num_pk          VARCHAR2(5)   NOT NULL,
        fecha_inicial   DATE           NOT NULL,
        fecha_final     DATE,
        num_mascota_fk1  NUMBER(6)     NOT NULL,
        id_encasillado_fk2 NUMBER(3)   NOT NULL,
CONSTRAINT reservacion_nummascota_fk FOREIGN KEY (num_mascota_fk1)
        REFERENCES mascota(num_pk),
CONSTRAINT reservacion_idencasillado_fk FOREIGN KEY (id_encasillado_fk2)
        REFERENCES encasillado(id_pk),
CONSTRAINT reservacion_num_pk PRIMARY KEY (num_pk));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER

```

```

SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla record nivel-3 *
REM *****
CREATE TABLE record (
    id_clinica_pk_fk1          NUMBER(4)    NOT NULL,
    num_mascota_pk_fk2        NUMBER(6)    NOT NULL,
CONSTRAINT record_id_clinica_pk_fk1 FOREIGN KEY (id_clinica_pk_fk1)
    REFERENCES clinica(id_pk),
CONSTRAINT record_num_mascota_pk_fk2 FOREIGN KEY (num_mascota_pk_fk2)
    REFERENCES mascota(num_pk),
CONSTRAINT record_id_pk PRIMARY KEY (id_clinica_pk_fk1, num_mascota_pk_fk2));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla mascota-comida nivel-3 *
REM *****
CREATE TABLE mascotacomida (
    num_mascota_pk_fk1          NUMBER(6)    NOT NULL,
    num_comida_pk_fk2          VARCHAR2(5)  NOT NULL,
CONSTRAINT mascotacomida_nummascota_pk_fk1 FOREIGN KEY (num_mascota_pk_fk1)
    REFERENCES mascota(num_pk),
CONSTRAINT mascotacomida_numcomida_fk FOREIGN KEY (num_comida_pk_fk2)
    REFERENCES comida(num_pk),
CONSTRAINT mascotacomida_num_pk PRIMARY KEY (num_mascota_pk_fk1,
num_comida_pk_fk2));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla mascota-objeto nivel-3 *
REM *****
CREATE TABLE mascotaobjeto (
    num_mascota_pk_fk1          NUMBER(6,0)  NOT NULL,
    id_objeto_pk_fk2           VARCHAR2(3)  NOT NULL,
CONSTRAINT mascotaobjeto_nummascota_pk_fk1 FOREIGN KEY (num_mascota_pk_fk1)
    REFERENCES mascota(num_pk),
CONSTRAINT mascotaobjeto_idobjeto_fk FOREIGN KEY (id_objeto_pk_fk2)
    REFERENCES objeto(id_pk),
CONSTRAINT mascotaobjeto_id_pk PRIMARY KEY (num_mascota_pk_fk1,
id_objeto_pk_fk2));
REM *****
REM * Las próximas dos instrucciones muestran el nombre del *
REM * usuario de la cuenta y la fecha del sistema. Esto es *
REM * requerido para crear o eliminar tablas y otros objetos. *
REM *****
REM
SHOW USER
SELECT systimestamp FROM dual;
REM *****
REM * Creacion tabla mascota-condicion nivel-3 *
REM *****
CREATE TABLE mascotacondicion (
    num_mascota_pk_fk1          NUMBER(6,0)  NOT NULL,

```



```
        id_condicion_pk_fk2                VARCHAR2(25) NOT NULL,
CONSTRAINT mascotacondicion_nummascota_pk_fk1 FOREIGN KEY (num_mascota_pk_fk1)
        REFERENCES mascota(num_pk),
CONSTRAINT mascotacondicion_idcondicion_fk FOREIGN KEY (id_condicion_pk_fk2)
        REFERENCES condicion(id_pk),
CONSTRAINT mascotacondicion_id_pk PRIMARY KEY (num_mascota_pk_fk1,
id_condicion_pk_fk2));
COMMIT;
```

V. Evidencia de corrida. Llenar la siguiente tabla (30 pts):

Código (copy/paste) (Aquí va el pedazo del código del script que crea la tabla. Que sea legible y mantenga la indentación	Nivel (1,2 o 3)	Imagen de la corrida (incluya Username y fecha y hora) por tabla se debe ver la respuesta de Oracle de que la tabla se creó
<p><b>Tabla 1)</b></p> <pre>CREATE TABLE direccion (     id_pk          VARCHAR2(4)    NOT NULL,     calle          VARCHAR2(10)   NOT NULL,     urbanizacion   VARCHAR2(25),     ciudad         VARCHAR2(15)   NOT NULL,     estado         CHAR(2)        NOT NULL,     codigo_postal  CHAR(5)        NOT NULL,     CONSTRAINT direccion_id_pk      PRIMARY KEY (id_pk));</pre>	1	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.138000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla direccion nivel-1 SQL&gt; REM ***** SQL&gt; CREATE TABLE direccion ( 2      id_pk          VARCHAR2(4)    NOT NULL, 3      calle          VARCHAR2(10)   NOT NULL, 4      urbanizacion   VARCHAR2(25), 5      ciudad         VARCHAR2(15)   NOT NULL, 6      estado         CHAR(2)        NOT NULL, 7      codigo_postal  CHAR(5)        NOT NULL, 8      CONSTRAINT direccion_id_pk      PRIMARY KEY (id_pk));  Table created.</pre>
<p><b>Tabla 2)</b></p> <pre>CREATE TABLE animal (     codigo_pk      VARCHAR2(3)    NOT NULL,     nombre         VARCHAR2(25)   NOT NULL,     descripcion    VARCHAR2(99)   NOT NULL,     CONSTRAINT animal_codigo_pk      PRIMARY KEY (codigo_pk));</pre>	1	<pre>SQL&gt; SHOW USER USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.198000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla animal nivel-1 SQL&gt; REM ***** SQL&gt; CREATE TABLE animal ( 2      codigo_pk      VARCHAR2(3)    NOT NULL, 3      nombre         VARCHAR2(25)   NOT NULL, 4      descripcion    VARCHAR2(99)   NOT NULL, 5      CONSTRAINT animal_codigo_pk      PRIMARY KEY (codigo_pk));  Table created.</pre>

Código	Nivel (1,2 o 3)	Imagen de la corrida (incluya <i>Username</i> , fecha y hora)
<p><b>Tabla 3)</b></p> <pre>CREATE TABLE objeto (     id_pk          VARCHAR2(2)  NOT NULL,     nombre         VARCHAR2(15) NOT NULL,     CONSTRAINT objeto_id_pk         PRIMARY KEY (id_pk));</pre>	<p><b>1</b></p>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.208000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla objeto nivel-1          * SQL&gt; REM ***** SQL&gt; CREATE TABLE objeto (     2          id_pk          VARCHAR2(2)  NOT NULL,     3          nombre         VARCHAR2(15) NOT NULL,     4  CONSTRAINT objeto_id_pk          PRIMARY KEY (id_pk));  Table created.</pre>
<p><b>Tabla 4)</b></p> <pre>CREATE TABLE condicion (     id_pk          VARCHAR2(5)  NOT NULL,     nombre         VARCHAR2(15) NOT NULL,     descripcion    VARCHAR2(99) NOT NULL,     CONSTRAINT condicion_id_pk         PRIMARY KEY (id_pk));</pre>	<p><b>1</b></p>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.234000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla condicion nivel-1          * SQL&gt; REM ***** SQL&gt; CREATE TABLE condicion (     2          id_pk          VARCHAR2(5)  NOT NULL,     3          nombre         VARCHAR2(15) NOT NULL,     4          descripcion    VARCHAR2(99) NOT NULL,     5  CONSTRAINT condicion_id_pk          PRIMARY KEY (id_pk));  Table created.</pre>

Código	Nivel (1,2 o 3)	Imagen de la corrida (incluya <i>Username</i> , fecha y hora)
<b>Tabla 5)</b>  <pre>CREATE TABLE encasillado (     id_pk          NUMBER(3)    NOT NULL,     descripcion    VARCHAR2(99) NOT NULL,     CONSTRAINT encasillado_id_pk PRIMARY KEY (id_pk));</pre>	<b>1</b>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.248000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla encasillado nivel-1          * SQL&gt; REM ***** SQL&gt; CREATE TABLE encasillado (     2      id_pk          NUMBER(3)    NOT NULL,     3      descripcion    VARCHAR2(99) NOT NULL,     4  CONSTRAINT encasillado_id_pk PRIMARY KEY (id_pk));  Table created.</pre>
<b>Tabla 6)</b>  <pre>CREATE TABLE dueno (     id_pk          NUMBER(5)    NOT NULL,     primer_nombre  VARCHAR2(20) NOT NULL,     inicial        CHAR(1),     apellido_paterno VARCHAR2(20) NOT NULL,     apellido_materno VARCHAR2(20),     celular        CHAR(10)    NOT NULL,     correo_electronico VARCHAR2(30) NOT NULL,     id_direccion_fk VARCHAR2(4)  NOT NULL,     CONSTRAINT dueno_iddireccion_fk FOREIGN KEY (id_direccion_fk)         REFERENCES direccion(id_pk),     CONSTRAINT dueno_id_pk PRIMARY KEY (id_pk));</pre>	<b>2</b>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.268000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla dueno nivel-2          * SQL&gt; REM ***** SQL&gt; CREATE TABLE dueno (     2      id_pk          NUMBER(5)    NOT NULL,     3      primer_nombre  VARCHAR2(20) NOT NULL,     4      inicial        CHAR(1),     5      apellido_paterno VARCHAR2(20) NOT NULL,     6      apellido_materno VARCHAR2(20),     7      celular        CHAR(10)    NOT NULL,     8      correo_electronico VARCHAR2(30) NOT NULL,     9      id_direccion_fk VARCHAR2(4)  NOT NULL,     10  CONSTRAINT dueno_iddireccion_fk FOREIGN KEY (id_direccion_fk)     11      REFERENCES direccion(id_pk),     12  CONSTRAINT dueno_id_pk PRIMARY KEY (id_pk));  Table created.</pre>

Código	Nivel (1,2 o 3)	Imagen de la corrida (incluya <i>Username</i> , fecha y hora)
<p><b>Tabla 7)</b></p> <pre>CREATE TABLE comida (     num_pk                VARCHAR2(5)    NOT NULL,     nombre                VARCHAR2(25)   NOT NULL,     unidad_medida         VARCHAR2(15)   NOT NULL,     id_dueno_fk            NUMBER(7,0)    NOT NULL, CONSTRAINT comida_iddueno_fk FOREIGN KEY (id_dueno_fk)     REFERENCES dueno(id_pk), CONSTRAINT comida_num_pk  PRIMARY KEY (num_pk));</pre>	2	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.289000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla comida nivel-2          * SQL&gt; REM ***** SQL&gt; CREATE TABLE comida (     2      num_pk                VARCHAR2(5)    NOT NULL,     3      nombre                VARCHAR2(25)   NOT NULL,     4      unidad_medida         VARCHAR2(15)   NOT NULL,     5      id_dueno_fk            NUMBER(7,0)    NOT NULL,     6  CONSTRAINT comida_iddueno_fk FOREIGN KEY (id_dueno_fk)     7                                REFERENCES dueno(id_pk),     8  CONSTRAINT comida_num_pk  PRIMARY KEY (num_pk));  Table created.</pre>
<p><b>Tabla 8)</b></p> <pre>CREATE TABLE oficina (     id_pk                NUMBER(4)        NOT NULL,     telefono             CHAR(10)         NOT NULL,     correo_electronico   VARCHAR2(30)     NOT NULL,     id_direccion_fk      VARCHAR2(4)      NOT NULL, CONSTRAINT oficina_iddireccion_fk FOREIGN KEY (id_direccion_fk)     REFERENCES direccion(id_pk), CONSTRAINT oficina_id_pk  PRIMARY KEY (id_pk));</pre>	2	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.298000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla oficina nivel-2          * SQL&gt; REM ***** SQL&gt; CREATE TABLE oficina (     2      id_pk                NUMBER(4)        NOT NULL,     3      telefono             CHAR(10)         NOT NULL,     4      correo_electronico   VARCHAR2(30)     NOT NULL,     5      id_direccion_fk      VARCHAR2(4)      NOT NULL,     6  CONSTRAINT oficina_iddireccion_fk FOREIGN KEY (id_direccion_fk)     7                                REFERENCES direccion(id_pk),     8  CONSTRAINT oficina_id_pk  PRIMARY KEY (id_pk));  Table created.</pre>

Código	Nivel (1,2 o 3)	Imagen de la corrida (incluya <i>Username</i> , fecha y hora)	
<p><b>Tabla 9)</b></p> <pre>CREATE TABLE clinica (     id_pk          NUMBER(4)      NOT NULL,     telefono       CHAR(10)       NOT NULL,     correo_electronico VARCHAR2(30) NOT NULL,     id_direccion_fk VARCHAR2(4)   NOT NULL,     CONSTRAINT clinica_iddireccion_fk FOREIGN KEY         (id_direccion_fk)             REFERENCES direccion(id_pk),     CONSTRAINT clinica_id_pk PRIMARY KEY (id_pk));</pre>	2	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.318000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla clinica nivel-2          * SQL&gt; REM ***** SQL&gt; CREATE TABLE clinica (     2      id_pk          NUMBER(4)      NOT NULL,     3      telefono       CHAR(10)       NOT NULL,     4      correo_electronico VARCHAR2(30) NOT NULL,     5      id_direccion_fk VARCHAR2(4)   NOT NULL,     6  CONSTRAINT clinica_iddireccion_fk FOREIGN KEY (id_direccion_fk)     7      REFERENCES direccion(id_pk),     8  CONSTRAINT clinica_id_pk PRIMARY KEY (id_pk));  Table created.</pre>	
<p><b>Tabla 10)</b></p> <pre>CREATE TABLE departamento (     codigo_pk      CHAR(2)      NOT NULL,     nombre         VARCHAR2(20) NOT NULL,     descripcion    VARCHAR2(99) NOT NULL,     id_oficina_fk  NUMBER(4)    NOT NULL,     CONSTRAINT departamento_idoficina_fk FOREIGN KEY         (id_oficina_fk)             REFERENCES oficina(id_pk),     CONSTRAINT departamento_codigo_pk PRIMARY KEY         (codigo_pk);CONSTRAINT departamento_codigo_pk     PRIMARY KEY (codigo_pk));</pre>	2	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.06.31.346000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla departamento nivel-2          * SQL&gt; REM ***** SQL&gt; CREATE TABLE departamento (     2      codigo_pk      CHAR(2)      NOT NULL,     3      nombre         VARCHAR2(20) NOT NULL,     4      descripcion    VARCHAR2(99) NOT NULL,     5      id_direccion_fk VARCHAR2(4)  NOT NULL,     6  CONSTRAINT departamento_iddireccion_fk FOREIGN KEY (id_direccion_fk)     7      REFERENCES direccion(id_pk),     8  CONSTRAINT departamento_codigo_pk PRIMARY KEY (codigo_pk));  Table created.</pre>	

Código	Nivel (1,2 o 3)	Imagen de la corrida (incluya <i>Username</i> , fecha y hora)
<p><b>Tabla 11)</b></p> <pre>CREATE TABLE empleado (     num_pk                NUMBER(5)          NOT NULL,     primer_nombre         VARCHAR2(20)        NOT NULL,     inicial               CHAR(1),     apellido_paterno      VARCHAR2(20)        NOT NULL,     apellido_materno      VARCHAR2(20),     tipo                  CHAR(1),     salario_mensual       NUMBER(7,2),     num_contrato           NUMBER(4),     tarifa_contrato        NUMBER(5,2),     salario_x_hora         NUMBER(4,2),     horas_trabajadas       NUMBER(2),     fecha_nacimiento       DATE                NOT NULL,     correo_electronico     VARCHAR2(30)        NOT NULL,     id_direccion_fk1       VARCHAR2(4)         NOT NULL,     codigo_departamento_fk2 CHAR(2),     id_clinica_fk3         NUMBER(4),     CHECK ((tipo) IN ('C', 'H', 'A')),     CONSTRAINT empleado_iddireccion_fk1 FOREIGN KEY         (id_direccion_fk1)             REFERENCES direccion(id_pk),     CONSTRAINT empleado_codigodepartamento_fk2 FOREIGN     KEY  (codigo_departamento_fk2)             REFERENCES departamento(codigo_pk),     CONSTRAINT empleado_idclinica_fk3 FOREIGN KEY         (id_clinica_fk3)             REFERENCES clinica(id_pk),     CONSTRAINT empleado_num_pk PRIMARY KEY         (num_pk));</pre>	<p><b>3</b></p>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.21.18.749000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla empleado nivel-3          * SQL&gt; REM ***** SQL&gt; CREATE TABLE empleado ( 2      num_pk                NUMBER(5)          NOT NULL, 3      primer_nombre         VARCHAR2(20)        NOT NULL, 4      inicial               CHAR(1), 5      apellido_paterno      VARCHAR2(20)        NOT NULL, 6      apellido_materno      VARCHAR2(20), 7      tipo                  CHAR(1), 8      salario_mensual       NUMBER(7,2), 9      num_contrato           NUMBER(4), 10     tarifa_contrato        NUMBER(5,2), 11     salario_x_hora         NUMBER(4,2), 12     horas_trabajadas       NUMBER(2), 13     fecha_nacimiento       DATE                NOT NULL, 14     correo_electronico     VARCHAR2(30)        NOT NULL, 15     id_direccion_fk1       VARCHAR2(4)         NOT NULL, 16     codigo_departamento_fk2 CHAR(2), 17     id_clinica_fk3         NUMBER(4), 18     CHECK ((tipo) IN ('C', 'H', 'A')), 19     CONSTRAINT empleado_iddireccion_fk1 FOREIGN KEY (id_direccion_fk1) 20         REFERENCES direccion(id_pk), 21     CONSTRAINT empleado_codigodepartamento_fk2 FOREIGN KEY (codigo_departamento_fk2) 22         REFERENCES departamento(codigo_pk), 23     CONSTRAINT empleado_idclinica_fk3 FOREIGN KEY (id_clinica_fk3) 24         REFERENCES clinica(id_pk), 25     CONSTRAINT empleado_num_pk PRIMARY KEY (num_pk));  Table created.</pre>

Tabla 12)

```
CREATE TABLE mascota (  
    num_pk                NUMBER(6)    NOT NULL,  
    nombre                VARCHAR2(15) NOT NULL,  
    fecha_nacimiento      DATE          NOT NULL,  
    raza                  VARCHAR2(20) NOT NULL,  
    id_dueno_fk1          NUMBER(7,0)  NOT NULL,  
    codigo_animal_fk2     VARCHAR2(4)  NOT NULL,  
    CONSTRAINT mascota_iddueno_fk FOREIGN KEY  
    (id_dueno_fk1)  
        EFERENCES dueno(id_pk),  
    CONSTRAINT mascota_codanimal_fk FOREIGN KEY  
    (codigo_animal_fk2)  
        REFERENCES animal(codigo_pk),  
    CONSTRAINT mascota_num_pk PRIMARY KEY (num_pk));
```

3

```
USER is "C##JALVAREZ6451"  
SQL> SELECT systimestamp FROM dual;  
  
SYSTIMESTAMP  
-----  
21-MAY-22 05.21.18.799000 PM -04:00  
  
SQL> REM *****  
SQL> REM *          Creacion tabla mascota nivel-3          *  
SQL> REM *****  
SQL> CREATE TABLE mascota (  
2      num_pk                NUMBER(6)    NOT NULL,  
3      nombre                VARCHAR2(15) NOT NULL,  
4      fecha_nacimiento      DATE          NOT NULL,  
5      raza                  VARCHAR2(20) NOT NULL,  
6      id_dueno_fk1          NUMBER(7,0)  NOT NULL,  
7      codigo_animal_fk2     VARCHAR2(4)  NOT NULL,  
8      CONSTRAINT mascota_iddueno_fk FOREIGN KEY (id_dueno_fk1)  
9      REFERENCES dueno(id_pk),  
10     CONSTRAINT mascota_codanimal_fk FOREIGN KEY (codigo_animal_fk2)  
11     REFERENCES animal(codigo_pk),  
12     CONSTRAINT mascota_num_pk PRIMARY KEY (num_pk));  
  
Table created.
```



Código	Nivel (1,2 o 3)	Imagen de la corrida (incluya <i>Username</i> , fecha y hora)	
<p><b>Tabla 13)</b></p> <pre>CREATE TABLE reservacion (     num_pk                VARCHAR2(5)  NOT NULL,     fecha_inicial         DATE          NOT NULL,     fecha_final           DATE,     num_mascota_fk1       NUMBER(6)     NOT NULL,     id_encasillado_fk2    NUMBER(3)     NOT NULL,     CONSTRAINT reservacion_nummascota_fk FOREIGN KEY (num_mascota_fk1)         REFERENCES mascota(num_pk),     CONSTRAINT reservacion_idencasillado_fk FOREIGN KEY (id_encasillado_fk2)         REFERENCES encasillado(id_pk),     CONSTRAINT reservacion_num_pk PRIMARY KEY (num_pk));</pre>	<p><b>3</b></p>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.21.19.139000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla reservacion nivel-3          * SQL&gt; REM ***** SQL&gt; CREATE TABLE reservacion (     2      num_pk                VARCHAR2(5)  NOT NULL,     3      fecha_inicial         DATE          NOT NULL,     4      fecha_final           DATE,     5      num_mascota_fk1       NUMBER(6)     NOT NULL,     6      id_encasillado_fk2    NUMBER(3)     NOT NULL,     7  CONSTRAINT reservacion_nummascota_fk FOREIGN KEY (num_mascota_fk1)     8      REFERENCES mascota(num_pk),     9  CONSTRAINT reservacion_idencasillado_fk FOREIGN KEY (id_encasillado_fk2)    10      REFERENCES encasillado(id_pk),    11  CONSTRAINT reservacion_num_pk PRIMARY KEY (num_pk));  Table created.</pre>	
<p><b>Tabla 14)</b></p> <pre>CREATE TABLE record (     id_clinica_pk_fk1      NUMBER(4)     NOT NULL,     num_mascota_pk_fk2     NUMBER(6)     NOT NULL,     CONSTRAINT record_id_clinica_pk_fk1 FOREIGN KEY (id_clinica_pk_fk1)         REFERENCES clinica(id_pk),     CONSTRAINT record_num_mascota_pk_fk2 FOREIGN KEY (num_mascota_pk_fk2)         REFERENCES mascota(num_pk),     CONSTRAINT record_id_pk PRIMARY KEY (id_clinica_pk_fk1, num_mascota_pk_fk2));</pre>	<p><b>3</b></p>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.21.19.259000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla record nivel-3          * SQL&gt; REM ***** SQL&gt; CREATE TABLE record (     2      id_clinica_pk_fk1      NUMBER(4)     NOT NULL,     3      num_mascota_pk_fk2     NUMBER(6)     NOT NULL,     4  CONSTRAINT record_id_clinica_pk_fk1 FOREIGN KEY (id_clinica_pk_fk1)     5      REFERENCES clinica(id_pk),     6  CONSTRAINT record_num_mascota_pk_fk2 FOREIGN KEY (num_mascota_pk_fk2)     7      REFERENCES mascota(num_pk),     8  CONSTRAINT record_id_pk PRIMARY KEY (id_clinica_pk_fk1, num_mascota_pk_fk2));  Table created.</pre>	

Código	Nivel (1,2 o 3)	Imagen de la corrida (incluya <i>Username</i> , fecha y hora)
<p><b>Tabla 15)</b></p> <pre>CREATE TABLE mascotacomida (     num_mascota_pk_fk1      NUMBER(6)      NOT NULL,     num_comida_pk_fk2       VARCHAR2(5)    NOT NULL, CONSTRAINT mascotacomida_nummascota_pk_fk1 FOREIGN KEY  (num_mascota_pk_fk1)             REFERENCES mascota(num_pk) , CONSTRAINT mascotacomida_numcomida_fk      FOREIGN KEY  (num_comida_pk_fk2)             REFERENCES comida(num_pk) , CONSTRAINT mascotacomida_num_pk      PRIMARY KEY (num_mascota_pk_fk1, num_comida_pk_fk2));</pre>	<p><b>3</b></p>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.21.19.439000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla mascota-comida nivel-3          * SQL&gt; REM ***** SQL&gt; CREATE TABLE mascotacomida (     2          num_mascota_pk_fk1          NUMBER(6)      NOT NULL,     3          num_comida_pk_fk2           VARCHAR2(5)    NOT NULL,     4  CONSTRAINT mascotacomida_nummascota_pk_fk1  FOREIGN KEY  (num_mascota_pk_fk1)     5   REFERENCES mascota(num_pk),     6  CONSTRAINT mascotacomida_numcomida_fk      FOREIGN KEY  (num_comida_pk_fk2)     7   REFERENCES comida(num_pk),     8  CONSTRAINT mascotacomida_num_pk      PRIMARY KEY  (num_mascota_pk_fk1, num_comida_pk_fk2));  Table created.</pre>
<p><b>Tabla 16)</b></p> <pre>CREATE TABLE mascotaobjeto (     num_mascota_pk_fk1      NUMBER(6,0)    NOT NULL,     id_objeto_pk_fk2       VARCHAR2(3)    NOT NULL, CONSTRAINT mascotaobjeto_nummascota_pk_fk1 FOREIGN KEY  (num_mascota_pk_fk1)             REFERENCES mascota(num_pk) , CONSTRAINT mascotaobjeto_idobjeto_fk      FOREIGN KEY  (id_objeto_pk_fk2)             REFERENCES objeto(id_pk) , CONSTRAINT mascotaobjeto_id_pk      PRIMARY KEY (num_mascota_pk_fk1, id_objeto_pk_fk2));</pre>	<p><b>3</b></p>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.21.19.636000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla mascota-objeto nivel-3          * SQL&gt; REM ***** SQL&gt; CREATE TABLE mascotaobjeto (     2          num_mascota_pk_fk1          NUMBER(6,0)    NOT NULL,     3          id_objeto_pk_fk2           VARCHAR2(3)    NOT NULL,     4  CONSTRAINT mascotaobjeto_nummascota_pk_fk1  FOREIGN KEY  (num_mascota_pk_fk1)     5   REFERENCES mascota(num_pk),     6  CONSTRAINT mascotaobjeto_idobjeto_fk      FOREIGN KEY  (id_objeto_pk_fk2)     7   REFERENCES objeto(id_pk),     8  CONSTRAINT mascotaobjeto_id_pk      PRIMARY KEY  (num_mascota_pk_fk1, id_objeto_pk_fk2));  Table created.</pre>

Código	Nivel (1,2 o 3)	Imagen de la corrida (incluya <i>Username</i> , fecha y hora)
<p><b>Tabla 17)</b></p> <pre>CREATE TABLE mascotacondicion (     num_mascota_pk_fk1          NUMBER(6,0)  NOT NULL,     id_condicion_pk_fk2        VARCHAR2(25) NOT NULL, CONSTRAINT mascotacondicion_nummascota_pk_fk1  FOREIGN KEY (num_mascota_pk_fk1)     REFERENCES mascota(num_pk) , CONSTRAINT mascotacondicion_idcondicion_fk      FOREIGN KEY (id_condicion_pk_fk2)     REFERENCES condicion(id_pk) , CONSTRAINT mascotacondicion_id_pk      PRIMARY KEY (num_mascota_pk_fk1, id_condicion_pk_fk2));</pre>	<p><b>3</b></p>	<pre>USER is "C##JALVAREZ6451" SQL&gt; SELECT systimestamp FROM dual;  SYSTIMESTAMP ----- 21-MAY-22 05.21.20.219000 PM -04:00  SQL&gt; REM ***** SQL&gt; REM *          Creacion tabla mascota-condicion nivel-3          * SQL&gt; REM ***** SQL&gt; CREATE TABLE mascotacondicion ( 2      num_mascota_pk_fk1          NUMBER(6,0)  NOT NULL, 3      id_condicion_pk_fk2        VARCHAR2(25) NOT NULL, 4  CONSTRAINT mascotacondicion_nummascota_pk_fk1      FOREIGN KEY  (num_mascota_pk_fk1) 5      REFERENCES mascota(num_pk), 6  CONSTRAINT mascotacondicion_idcondicion_fk  FOREIGN KEY  (id_condicion_pk_fk2) 7      REFERENCES condicion(id_pk), 8  CONSTRAINT mascotacondicion_id_pk      PRIMARY KEY  (num_mascota_pk_fk1, id_condicion_pk_fk2));  Table created.</pre>